

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství



Konfigurovatelný vypínač pro domovní elektroinstalaci
A Configurable Switch for House Wiring

2016

Jan Velička

Zadání bakalářské práce

Student: **Jan Velička**

Studijní program: B2649 Elektrotechnika

Studijní obor: 2612R041 Řídicí a informační systémy

Téma: Konfigurovatelný vypínač pro domovní elektroinstalaci
A Configurable Switch for House Wiring

Jazyk vypracování: čeština

Zásady pro vypracování:

1. Zpracování přehledu Smart-Home technologií.
2. Návrh hardwaru pro konfigurovatelný vypínač.
3. Realizace a testování hardwaru.
4. Koncepční návrh softwaru zařízení.
5. Implementace softwaru do mikrokontroléru.
6. Testování řešení a zhodnocení výsledků práce.

Seznam doporučené odborné literatury:

- [1] HARPER, Richard. *Inside the smart home*. New York: Springer, c2003, xi, 264 p. ISBN 1852336889.
[2] WAGNER, Ferdinand. *Modeling software with finite state machines: a practical approach*. Boca Raton, FL: Auerbach, 2006, xix, 369 p. ISBN 978-0849380860.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Michal Prauzek, Ph.D.**

Datum zadání: 01.09.2015

Datum odevzdání: 29.04.2016



doc. Ing. Jiří Koziolek, Ph.D.
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 29. dubna 2016



.....
podpis studenta

Poděkování

Děkuji za rady a odbornou pomoc při vypracování bakalářské práce svému vedoucímu Ing. Michalu Prauzkovi, Ph.D. a také za konzultace a rady při řešení této práce Ing. Jaromíru Konečnému, Ph.D.

Abstrakt

Hlavní část této práce se zabývá návrhem a řešením konfigurovatelného časového spínače, který bude možné zabudovat do stávající elektroinstalace. Uživatel si může snadno vybírat časy pro časovač pomocí jediného tlačítka. Řešení je směřováno na použití mikrokontroléru ve spojení s elektronikou ve třech napěťových hladinách a s co nejmenší vlastní spotřebou elektrické energie. Je možné spínat obecnou zátěž pomocí výstupního relé. Práce se také zabývá porovnáním výrobků s podobnými vlastnostmi, které lze dnes na trhu pořídit. Práci doplňuje přehled systémů Smart-Home, které jsou vhodné pro dodatečnou instalaci. Jedná se převážně o bezdrátová řešení.

Klíčová slova

Elektroinstalace, Mikrokontrolér, Konfigurovatelný, Spínač, Zátěž, Snadné ovládání, Smart-Home

Abstract

The main part of this thesis deals with configurable timer, which can be embedded into existing electrical wiring. The user can easily choose the times for the timer using a single button. Solution is routed to the use of the microcontroller in conjunction with the electronics in the three voltage levels and with the smallest possible own energy consumption. It is possible to switch a general load using a relay output. The thesis also presents a comparison of products that can be purchased on the market today with similar characteristics. Work complements the system overview Smart-Home, which are suitable for retrofitting. These are primarily wireless solutions.

Keywords

Wiring, Microcontroller, Switch, Load, Configurable, Easy usage, Smart-Home

Obsah

OBSAH	6
SEZNAM POUŽITÝCH SYMBOLŮ	7
SEZNAM POUŽITÝCH ZKRATEK	8
SEZNAM ILUSTRACÍ A SEZNAM TABULEK	9
1 ÚVOD	10
2 PŘEHLED	11
2.1 PŘEHLED A SROVNÁNÍ PODOBNÝCH ZAŘÍZENÍ	11
2.1.1 Schodišťový automat ELKO EP CRM-4	11
2.1.2 Programovatelný schodišťový automat ELKO EP CRM-42F	13
2.1.3 Schodišťový časový spínač	15
2.1.4 Konfigurovatelný časový spínač – navrženo v rámci této bakalářské práce	16
2.2 PŘEHLED SMART-HOME TECHNOLOGIÍ	17
2.2.1 Loxone MINISERVER GO Air	17
2.2.2 iNELS Smart Home	19
2.2.3 Zhodnocení	20
3 NÁVRH HARDWARU KONFIGUROVATELNÉHO SPÍNAČE	21
3.1 KONSTRUKCE	21
3.2 NAPÁJECÍ PŘEDŘADNÍK	22
3.3 STABILIZACE NAPĚTÍ	22
3.4 OVLÁDÁNÍ	23
3.5 SPÍNÁNÍ RELÉ	24
3.6 MIKROKONTROLÉR (MCU)	25
3.6.1 Volba vhodného kontroléru	25
3.6.2 Využití portů MCU	25
3.7 RGB LED	26
3.8 PROGRAMOVÁNÍ POMOCÍ ICSP	27
3.9 DESKA PLOŠNÝCH SPOJŮ	28
4 TESTOVÁNÍ A MĚŘENÍ HARDWARU	29
4.1 MĚŘENÍ	29
4.2 TESTOVÁNÍ	31
4.3 OPRAVY	32
4.3.1 Snížení činných ztrát	32
4.3.2 Úprava propojení LDO stabilizátoru	32
4.3.3 Zprovoznění nastavovacího mikro-tlačítka	32
5 NÁVRH SOFTWARE	33
5.1 POPIS FUNKCIONALITY	33
5.2 KONCEPČNÍ NÁVRH	34
5.3 STAVOVÝ DIAGRAM	36
6 IMPLEMENTACE DO MIKROKONTROLÉRU	37
6.1 POPIS FUNKCÍ	37
7 TESTOVÁNÍ CELÉHO ZAŘÍZENÍ	41
8 ZÁVĚR	43
LITERATURA	45
SEZNAM PŘÍLOH	46

Seznam použitých symbolů

Symbol	Jednotky	Význam symbolu
A	Wh	Energie
C	F	Kapacita
$\cos\varphi$	-	Účíník
f	Hz	Frekvence
I	A	Proud
L	H	Indukčnost
P	W	Činný výkon
PF	-	Power Factor, Celkový účíník
Q	var	Jalový výkon
R	Ω	Odpor
S	VA	Zdánlivý výkon
U	V	Napětí
Z	Ω	Impedance
φ	$^{\circ}$	Fázový posun - úhel mezi vektory napětí a proudu

Seznam použitých zkratek

Zkratka	Význam
DIL	Kontakty ve dvou řadách (Dual In Line)
DPS	Deska plošných spojů
ICSP	Sériové programování přímo v obvodu (In Circuit Serial Programming)
IP	Stupeň elektrického krytí (International Protection)
KNX	Celosvětový standart pro řízení budov(Konnex)
LDO	Nízko ztrátový (Low Drop Out)
MCLR	Externí reset (Master Clear pin)
MCU	Mikrokontrolér (Micro Controller Unit)
$n_{\text{přerušení}}$	Hodnota počtu přerušení za sekundu
$n_{\text{přerušení_B1}}$	Hodnota počtu přerušení pro barvu 1
$n_{\text{přerušení_B2}}$	Hodnota počtu přerušení pro barvu 2
$n_{\text{přerušení_perioda}}$	Hodnota počtu přerušení za jednu periodu bliknutí
RGB LED	Tříbarevná LED dioda (červená, zelená, modrá)
SMD	Součástka montována povrchově (Surface Mount Device)
SMT	Technologie povrchového osazování (Surface Mount Technology)
THD	Součástka montovaná přes otvor (Through Hole Device)
UML	Grafický jazyk pro vizualizaci (Unified Modeling Language)

Seznam ilustrací a seznam tabulek

Obr. 1: zapojení CRM-4 [1] a) tří-vodičové, b) čtyř-vodičové	12
Obr. 2: Modul ELKO EP CRM-4 [1].....	12
Obr. 3: Zapojení CRM-42F [2] a) tří-vodičové, b) čtyř-vodičové	14
Obr. 4: Modul ELKO EP CRM-42F [2]	14
Obr. 5: Zapojení schodišťového časového spínače	15
Obr. 6: Zapojení konfigurovatelného spínače	16
Obr. 7: Topologie systému Loxone [4]	18
Obr. 8: iNels "Chytrá RF krabice" [5].....	19
Obr. 9: Blokové schéma zařízení	21
Obr. 10: Umístění zařízení vzhledem k tlačítku ABB – Tango a KU68, pohled shora	21
Obr. 11: Zapojení napájecího předřadníku.....	22
Obr. 12: Zapojení stabilizace napájení	23
Obr. 13: Zapojení obvodu pro ovládání	23
Obr. 14: Zapojení vstupu pro nastavovací mikro-tlačítko.....	24
Obr. 15: Zapojení relé spínané pomocí MCU	24
Obr. 16: Rozložení pinů MCU PIC12F1840 [7]	25
Obr. 17: Zapojení portů MCU.....	25
Obr. 18: Zapojení rezistorů a RGB LED.....	26
Obr. 19: Standardní zapojení programátoru pomocí ICSP (zdroj: Microchip, upraveno)	27
Obr. 20: Programovací konektor ICSP	27
Obr. 21: Předloha DPS strana a) Horní (Top), b) Dolní (Botom).....	28
Obr. 22: Schéma zapojení pro měření	29
Obr. 23: Graf závislosti proudu na napájecím napětí	30
Obr. 24: Graf výkonové charakteristiky.....	30
Obr. 25: Předloha vývojové DPS	31
Obr. 26: Vývojové a testovací pracoviště	31
Obr. 27: Oprava zapojení LDO stabilizátoru	32
Obr. 28: Oprava zapojení vstupního obvodu tlačítka	32
Obr. 29: Koncepční schéma šesti-časového režimu	34
Obr. 30: Koncepční schéma pro dvou-časový režim	35
Obr. 31: Koncepční schéma pro nastavování režimu	35
Obr. 32: Diagram stavového automatu	36
Obr. 33: Ukázka kompletace zařízení	41
Obr. 34: Demonstrační modul konfigurovatelného spínače.....	41
Obr. 35: Diagram časových sad	42
Tab. 1: Tabulka naměřených hodnot.....	29
Tab. 2: Měření závislosti na stavu relé.....	31
Tab. 3: Parametry při spodním limitu napájecího napětí	31
Tab. 4: Hodnoty časových sad - hodnoty v paměti MCU	42

1 Úvod

Cílem této práce je návrh a realizace autonomního konfigurovatelného časového spínače pro montáž do stávající elektroinstalační krabice KU68 (hloubka 43 mm), jako náhrada za standardní spínač. Impulzem k návrhu tohoto zařízení byly časté situace jako například zapomenuté rozsvícené světlo nebo stále běžící ventilátor. Toto se může stát v místnostech s malým krátkodobým pohybem lidí, jako jsou komora, sklep, půda atd. Na trhu jsou dostupné pouze časové spínače s možností předem nastaveného času vypnutí, a protože tyto výrobky neumožňují operativně (bez rozebrání) čas měnit, bylo nutno takové zařízení vyvinout. Jedná se tedy o zařízení, u kterého uživatel může snadno měnit nastavený čas při běžném užívání.

Práce je rozdělena do následujících částí:

- První část se zabývá přehledem podobných zařízení, která lze dnes pořídit a jak tato zařízení fungují. Případně jejich výhodami a nevýhodami.
- Druhá část se zabývá návrhem elektrických i mechanických částí konfigurovatelného spínače, jeho vývojem a optimalizací.
- Třetí část popisuje, jak probíhalo testování a měření parametrů zařízení. Jaké parametry zařízení byly sledovány
- Čtvrtá část práce řeší návrh softwaru včetně koncepce a ULM diagramů.
- Pátá část práce se zabývá implementací softwaru do jazyka C pro 8bitové mikrokontroléry Microchip.
- Šestá část popisuje testování celého zařízení.
- Sedmá část obsahuje zhodnocení výsledků této bakalářské práce.

2 Přehled

2.1 Přehled a srovnání podobných zařízení

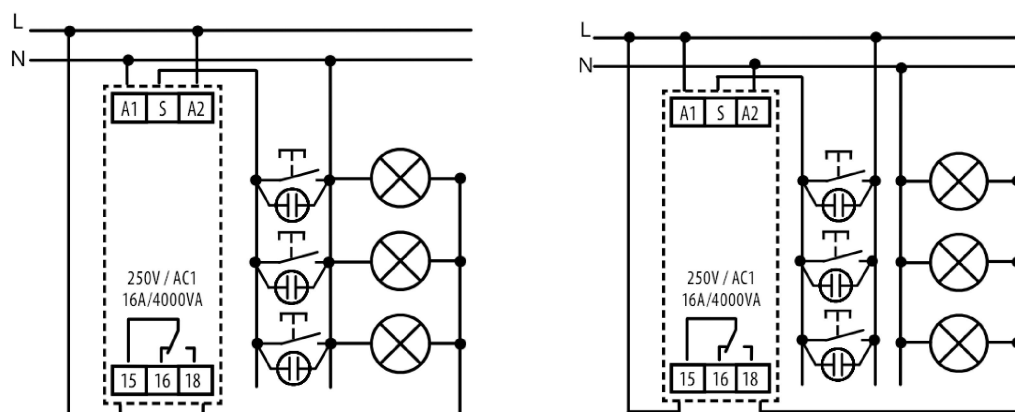
Tato kapitola se zabývá přehledem a srovnáním podobných zařízení, poříditelných na trhu. Jedná se o zařízení, která je možné dodatečně namontovat, případně jimi nahradit stávající zařízení. Všechna srovnávaná zařízení upravují dobu sepnutí obvodu, do kterého jsou zapojena. Tato zařízení však upravují dobu vypnutí ještě před běžným užíváním. Většinou se jedná o jediný nastavený čas vypnutí nebo násobky základního času.

2.1.1 Schodišťový automat ELKO EP CRM-4

Tento modul je určen pro montáž na DIN lištu, plní funkci schodišťového automatu. Pomocí potenciometru nastavujeme dobu časovače, po kterou je sepnuto relé. Posuvným přepínačem se volí režim fungování zařízení. Ovládání je pomocí až 35 paralelně pospojovaných tlačítek. Výstup je řešen kontaktem relé, může spínat odporovou, induktivní i kapacitní zátěž. Stav výstupu je signalizován pomocí červené LED. Svorky A1, A2 slouží pro připojení napájení. Dle připojení napájecích svorek k vodičům L a N, je použito tří-vodičové nebo čtyř-vodičové zapojení viz. Obr. 1. Svorka S je určena pro připojení tlačítek a dle použitého zapojení spínají tlačítka proti L nebo N vodiči. Tří-vodičové zapojení (Obr. 1a) se používá v případě, že spotřebiče a ovládací tlačítka mají společný N vodič. Čtyř-vodičové zapojení (Obr. 1b) se používá v případě, že tlačítka spínají proti L vodiči, který nemají společný. [1]

ELKO EP CRM-4 [1] má následující vlastnosti:

- Výstupní kontakt relé: 16 A
- Posuvný přepínač:
 - ON - trvale zapnuto (např. při úklidu, servisu)
 - AUTO - normální funkce dle nastaveného času
 - OFF - trvale vypnuto (např. při výměně žárovek)
- Časový rozsah: 30 s – 10 min
- Napájení: 230 V
- Spotřeba: 1,8 W
- Zátěž: RLC
- Ochrana proti zablokování tlačítka (zápalka v tlačítku)
- 1 modul DIN lišty
- Cena obvykle: 465,- Kč včetně DPH



Obr. 1 Zapojení CRM-4 [1] a) tří-vodičové, b) čtyř-vodičové



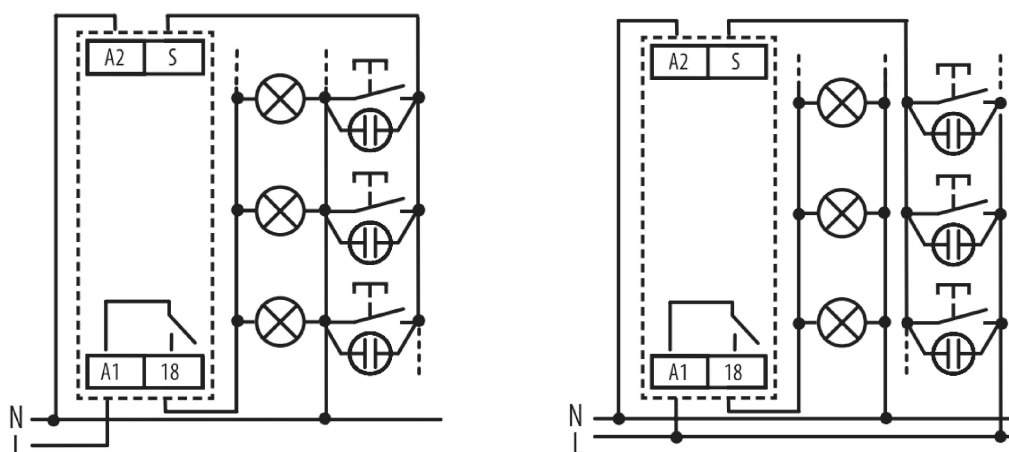
Obr. 2 Modul ELKO EP CRM-4 [1]

2.1.2 Programovatelný schodišťový automat ELKO EP CRM-42F

Tento modul je určen pro montáž na DIN lištu a plní funkci schodišťového automatu. Pomocí potenciometru se nastavuje doba časovače, po kterou je sepnuto relé. Posuvným přepínačem se volí režim fungování zařízení. Ovládání je pomocí až 100 paralelně pospojovaných tlačítek. Výstup je řešen kontaktem relé, může spínat odporovou, induktivní i kapacitní zátěž. Stav výstupu je signalizován pomocí červené LED, v režimu PROG a AUTO signalizuje sepnutí relé blikáním LED. Svorky A1, A2 slouží pro připojení napájení. Dle připojení napájecích svorek k vodičům L a N, je použito tří-vodičové (Obr. 3a) nebo čtyř-vodičové (Obr. 3b) zapojení. Svorka S je určena pro připojení tlačítek, dle použitého zapojení spínají tlačítka proti L nebo N vodiči. Při opakovaném krátkém stisknutí tlačítka dochází k násobení délky časování (např.: 3 stisky $\sim 3 \times 5$ min). Přidržením tlačítka na cca 2 s dojde k rozpojení kontaktu relé. [2]

ELKO EP CRM-42F [2] má následující vlastnosti:

- Výstupní kontakt relé: 16 A
- Posuvný přepínač
 - ON - výstup je trvale sepnutý, např. servisní režim
 - AUTO - časování dle nastavení potenciometru v rozsahu 30 s – 10 min
 - PROG - časování s možností prodloužení délky sepnutí počtem stisků tlačítka
- Napájecí napětí: 230 V
- Spotřeba: 1,8 W
- Zátěž: RLC
- 1 modul DIN lišty
- Cena obvykle: 690,- Kč včetně DPH



Obr. 3 zapojení CRM-42F [2] a) tří-vodičové, b) čtyř-vodičové



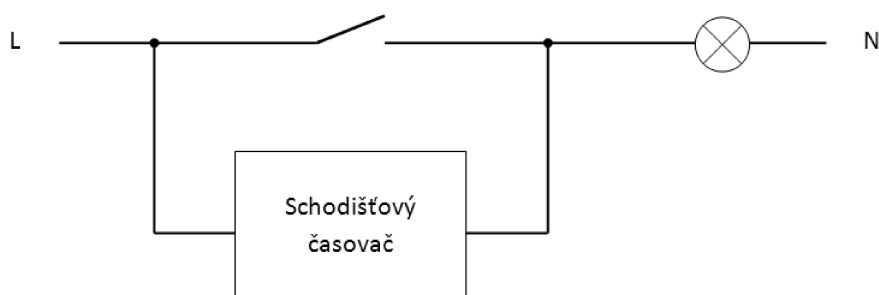
Obr. 4 Modul ELKO EP CRM-42F [2]

2.1.3 Schodišťový časový spínač

Jedná se o zařízení, jehož schéma bylo otištěné v Praktické elektronice č. 11/99 [3], umožňující opožděné zhasnutí. Zařízení se připojuje paralelně ke spínači. Připojuje se pouze dvěma vodiči do stávající elektroinstalace. Časovač se spouští při poklesu napětí, proto není vhodný do instalací, kde dochází k připojování výkonných spotřebičů. Zajímavou vlastností je způsob napájení, které nevyžaduje nulový vodič. Zapojení do elektroinstalace je uvedeno na obrázku (Obr. 5). V klidovém stavu je zařízení napájeno jalovým proudem. Spínání zajišťuje polovodičový prvek – triak. Z tohoto důvodu je možné spínat pouze odporovou zátěž. Stav je signalizován LED. [3]

Schodišťový časový spínač [3] má následující vlastnosti:

- Spínaný proud: 40 - 400 mA
- Časový rozsah: 0 s - několik min
- Napájecí napětí: 230 V
- Spotřeba: < 0,1 W
- Zátěž: R
- DPS do elektroinstalační krabice
- Cena: neuvedena



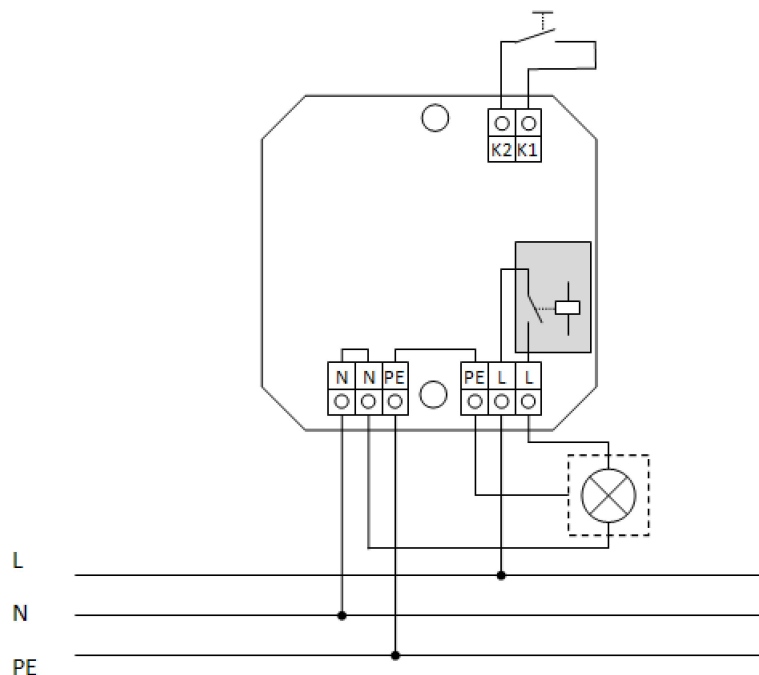
Obr. 5 Zapojení schodišťového časového spínače

2.1.4 Konfigurovatelný časový spínač – navrženo v rámci této bakalářské práce

Zařízení se montuje přímo do elektroinstalační krabice KU68 (hloubka 43 mm), pod elektroinstalační tlačítko ABB. Při stisku tlačítka se postupně mění barva RGB LED, blikáním dané barvy je signalizována zbývající doba časovače. Krátkým stiskem dojde k rozpojení kontaktu relé. Dlouhým stiskem dochází ke změně nastavení doby časovače.

Konfigurovatelný spínač má následující vlastnosti:

- Výstupní kontakt relé: 16 A
- Časový rozsah: 1 min - 255 h
- Napájecí napětí: 230 V
- Spotřeba: 0,3 W
- Zátěž: RLC
- Modul do elektroinstalační krabice KU68
- Cena při případné malosériové výrobě: 300,- Kč včetně DPH



Obr. 6 Zapojení konfigurovatelného spínače

2.2 Přehled Smart-Home technologií

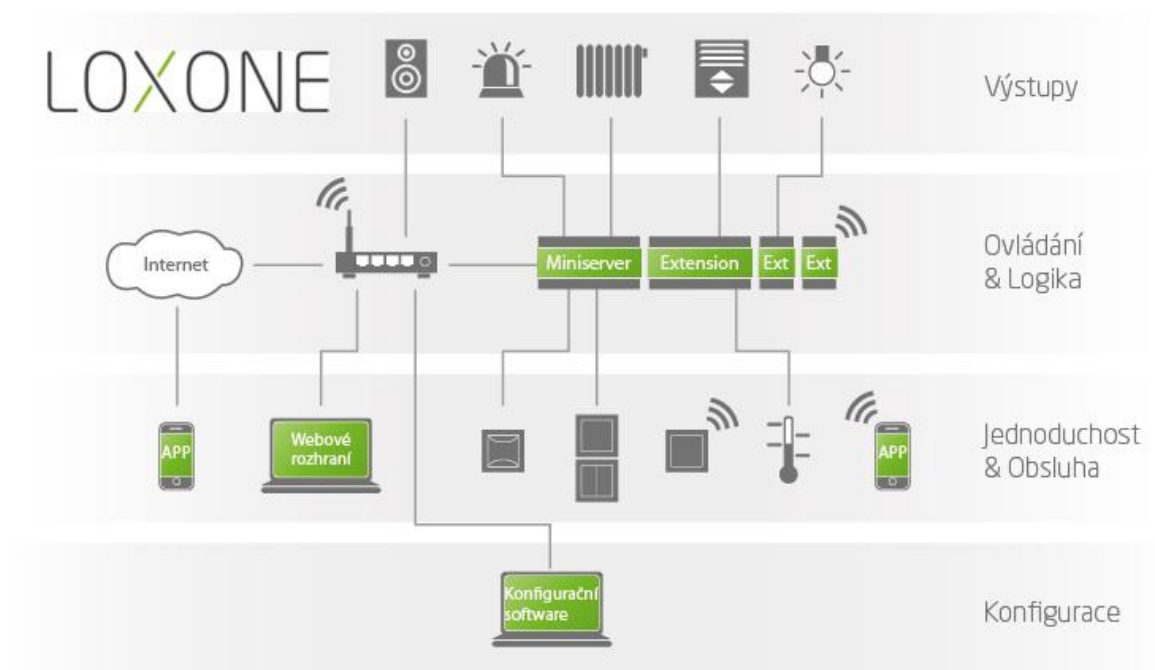
Kapitola se zabývá možnostmi a srovnáním Smart-Home technologií, které je možné si dnes pořídit do moderní domácnosti. Vybrané varianty jsou vhodné pro dodatečnou montáž, použití ve stávající elektroinstalaci. Využívají především bezdrátové prvky. Způsob řešení dodatečnou montáží byl zvolen z důvodu srovnatelnosti se zařízením popisovaným v této práci. Jedná se o systémy pro objekty menšího rozsahu, jako jsou byty a rodinné domy. V těchto objektech jsou systémy řízení prostředí většinou centralizované.

2.2.1 Loxone MINISERVER GO Air

Zařízení je řídicí jednotkou systému Loxone. Tento miniserver má možnost komunikovat s ostatními prvky bezdrátově i pomocí dvou-vodičové (diferenční) Loxone Bus. Tento miniserver, jak výrobce uvádí, je vhodný pro dodatečnou automatizaci domácnosti. Připojení do domácí sítě LAN je možné pomocí konektoru RJ-45. [4]

Specifikace dle katalogu:

- Napájení: 5 V (Micro USB)
- Kompaktní design pro montáž na stěnu
- Rozměry: 90 × 90 × 20 mm
- Operační systém Loxone OS s vestavěným webovým serverem
- Integrovaný Loxone Air Base Extension s interní anténou
- Frekvence: 868 MHz (SRD Band Europe) / 915 MHz (ISM Band Region 2)
- Konfigurační software Loxone Config
- Software pro ovládání z PC, prohlížeče a mobilních zařízení
- Možnost LAN připojení do sítě
- Slot pro Micro-SD (až 16 GB)
- Spotřeba: cca. 1,3 W
- Loxone Bus
- Stupeň krytí: IP20
- Provozní teplota: 0 až +50°C
- Maximální relativní vlhkost 95 % (nekondenzující)
- Cena je závislá na rozsahu instalace



Obr. 7 Topologie systému Loxone [4]

2.2.1.1 Shrnutí systému Loxone

Tento miniserver je vhodný pro dodatečnou automatizaci. Má široké možnosti příslušenství jako stmívače, termohlavice a další. Jedná se o centralizovaný řídicí systém. Výhodou je snadná instalace do domácnosti, ale pořád se jedná o systém poměrně nákladný. Nevýhodou je nutnost použití rozšiřujícího modulu pro drátové I/O. Výhodou je možnost použití KNX.

2.2.2 iNELS Smart Home

Jedná se o Smart-Home systém firmy ELKO EP. Je vhodný jak pro novostavby, tak i pro dodatečnou vestavbu do elektroinstalace. Většina přístrojů mezi sebou komunikuje bezdrátově na kmitočtu 868 MHz, obousměrně z důvodu zpětné vazby. Systém je připravený pro ovládání pomocí chytrého telefonu či tabletu, ale není to podmínkou. K dispozici jsou spínače, ovladače, stmívače, regulátory teploty a další. Tento systém v případě použití „Chytré RF krabičky“ má možnost připojení do domácí sítě LAN pomocí WIFI nebo Ethernet (RJ-45). Tato „Chytrá RF krabička“ je miniserverem tohoto systému. [5]

Specifikace dle katalogu:

- Napájení: 10-27 V nebo USB B
- Kompaktní design
- Rozměry: 90 × 52 × 65 mm
- Možnost připojení až 40 zařízení
- Frekvence: 868 MHz / 915 MHz / 916 MHz
- Software pro ovládání z prohlížeče a mobilních zařízení
- Možnost LAN připojení do sítě
- Možnost WIFI připojení do sítě
- Dosah: až 100 m (ve volném terénu)
- Spotřeba: max. 2 W
- Obousměrným protokol iNELS RF Control kompatibilní s Jablotron OASIS
- Stupeň krytí: IP20
- Provozní teplota: -20 až +50°C
- Maximální relativní vlhkost 95% (nekondenzující)
- Cena dle rozsahu instalace



Obr. 8 iNels "Chytrá RF krabička" [5]

2.2.2.1 Shrnutí systému iNels

Tento systém je celkově velice komplexně zpracovaný. Jedná se o centralizovaný řídicí systém. Jsou k dispozici moduly vhodné pro většinu zapojení, které se v domácnosti vyskytují. V nabídce je spínací/stmívací modul do elektroinstalační krabice nebo spínatelná/stmívatelná zásuvka, termohlavice pro regulaci teploty v místnosti, které bezdrátově komunikují s regulátorem teploty. Celkově je tento systém dobře a komplexně propracován tak, aby uživatel mohl mít pod kontrolou celý dům z chytrého telefonu či tabletu. Není to však podmínkou, je možné použít i hardwarové komponenty (tlačítka, dotykové panely ...). Cena je oproti systému Loxone nižší.

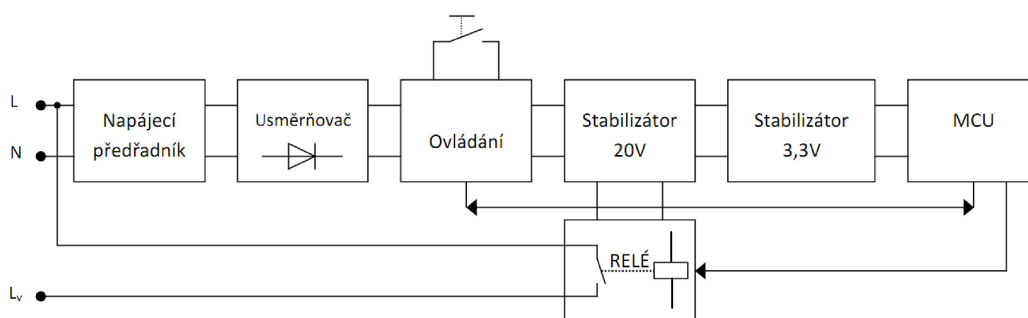
2.2.3 Zhodnocení

Oba popisované systémy jsou vhodné pro rodinné domy nebo byty. Oba systémy umožňují řízení prostředí v domě i bytě, jak osvětlení, tak teplotu. Popisované systémy jsou vhodné pro dodatečnou montáž do domu nebo bytu vzhledem k bezdrátové topologii. Systém iNels má mnohem větší spektrum zařízení, která lze použít.

3 Návrh hardwaru konfigurovatelného spínače

Charakteristické pro navrhované zařízení je ovládání jedním tlačítkem, autonomnost, kompaktnost a nízká vlastní spotřeba. Konfigurovatelný spínač umožňuje snadnou vizuální kontrolu pomocí signalizační RGB LED. Zařízení je navrženo tak, aby bylo univerzální a snadno implementovatelné do stávající elektroinstalace. Na základě těchto požadavků byly zvoleny návrhové principy, které byly použity při návrhu zařízení.

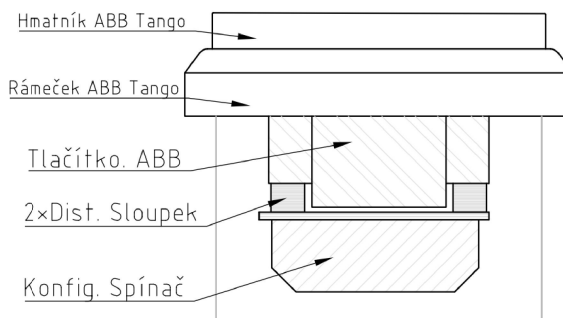
Zařízení je tvořeno funkčními bloky, které jsou vzájemně propojeny, jak je vidět na blokovém schématu (Obr. 9). Funkce a konfigurace jednotlivých bloků je popsána níže. Celkové zapojení jednotlivých bloků je uvedeno v příloze A.



Obr. 9 Blokové schéma zařízení

3.1 Konstrukce

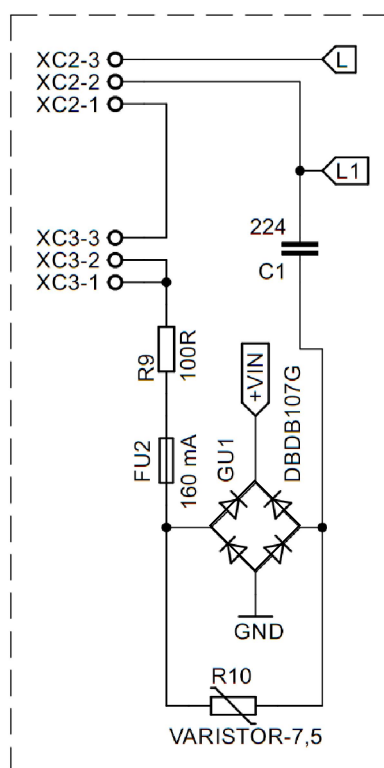
Zařízení je přizpůsobené pro montáž do standardní elektroinstalační krabice KU68 (hloubka 43 mm). Deska plošných spojů (DPS) je uzpůsobena pro montáž pod tlačítko ABB (Obr. 1), kam se přišroubuje (šroub M5 × 16 mm) přes distanční sloupky (6 mm). Rozteč děr se závity na tlačítku je 38 mm. DPS je navržena tak, aby ze strany, která přiléhá k tlačítku, bylo co nejméně součástí. To by zvětšovalo mezeru a tím i celkovou montážní hloubku. Na této straně DPS se nachází pouze RGB LED a nastavovací tlačítko. Připojení vodičů je do šroubovacích svorkovnic na DPS. V tlačítku ABB je otvor pro případnou montáž doutnavky, který je využit pro RGB LED. Instalace spínače musí být v souladu s normou ČSN EN 60 447 (Elektrotechnické předpisy Styk člověk – stroj Zásady pro ovládání) tak, aby stisk byl prováděn vždy v horní části hmatníku tlačítka. Výkresová dokumentace je uvedena v Příloze B.



Obr. 10 Umístění zařízení vzhledem k tlačítku ABB – Tango a KU68, pohled shora

3.2 Napájecí předřadník

Pro napájení zařízení bylo zvoleno řešení, které odpovídá druhu a účelu zařízení. V tomto případě to byly rozměry, jednoduchost a vlastní spotřeba elektrické energie. Těmto předpokladům nejlépe odpovídá kapacitní předřadník. Kapacitní předřadník je tvořen foliovým kondenzátorem 220 nF/250 V~. Rezistor v tomto zapojení vytváří samostatnou zem pro slaboproudou část, což je výhodné z hlediska stability a také bezpečnosti obvodu. Dojde-li v případě poruchy ke spojení obvodu s ochranným vodičem v síti, zareaguje na toto proudový chránič. Do napájecí větve je také zapojena 160 mA pojistka. Důvodem je jistění obvodu při poruše, zejména v případě poruchy vstupního předřadníku. Pro ochranu proti přepětí je použit varistor připojený mezi fázový a nulový vodič. Případné přepětí je svedeno do nulového vodiče. Pro napájení dalších obvodů je zapotřebí napětí usměrnit. Toto zajišťuje integrovaný můstkový 1A usměrňovač. Napájecí předřadník je realizován THD součástkami.

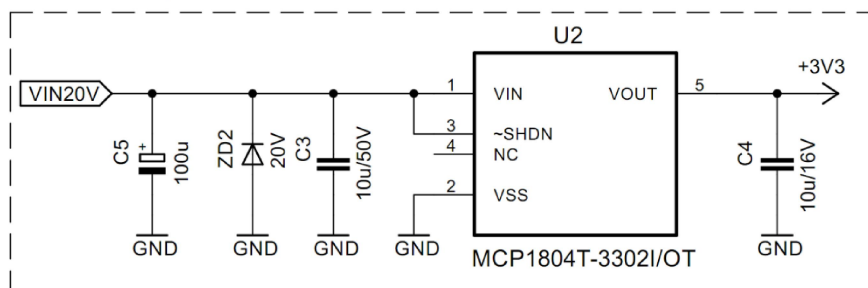


Obr. 11 Zapojení napájecího předřadníku

3.3 Stabilizace napětí

Stabilizace napětí zde byla řešena v několika napěťových úrovních. Důvodem je to, že jednotlivé funkční bloky zařízení vyžadují, aby byly napájeny rozdílným napětím. Pro spínání kontaktu relé je zapotřebí napětí 18-24V=. V tomto případě bylo zvoleno napětí 20V=. Důvodem bylo snížení výkonové ztráty na cívce relé. Pro stabilizaci na hodnotě 20 V= byla použita Zenerova dioda. Pro napájení mikrokontroléru bylo zvoleno napětí 3,3 V=, které je také výhodné pro předřadné odpory u LED. Jako stabilizátor napětí zde byl využit integrovaný stabilizátor napětí MCP1804T, vyráběný firmou Microchip Inc. Jedná se o LDO stabilizátor, který vykazuje nízké ztráty naprázdno (klidový proud je 50 μ A, jak je uvedeno v katalogovém listu [6]). Byly zde také zapojeny tři kondenzátory, dle

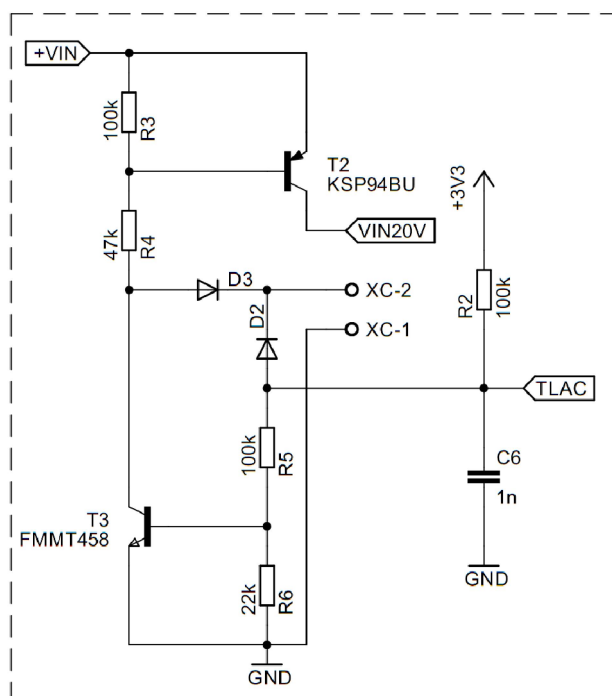
katalogového listu [6]. Elektrolytický kondenzátor C5/100 μ F slouží jako vyhlazovací kondenzátor a také pro eliminaci poklesu napětí při spínacím špičkovém proudu. Toto vzniká při přechodovém jevu při spínání cívky relé. Kondenzátory C3/1 μ F a C4/1 μ F byly zapojeny dle katalogového listu [6] integrovaného stabilizátoru jako filtrační.



Obr. 12 Zapojení stabilizace napájení

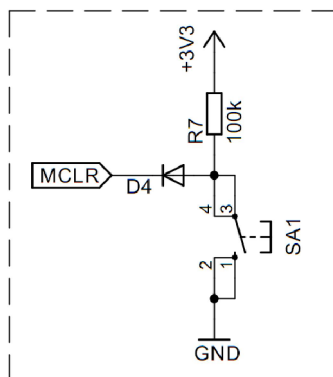
3.4 Ovládání

Pro automatické odpojení od zdroje bylo využito zapojení, které se používá v některých zařízeních napájených bateriemi. Toto zapojení funguje jako samo-přidrzný stykač. Při stisknutí tlačítka je připojen odporový dělič k zemi a otevřen tranzistor T2, následně se na pinu mikrokontroléru objeví log. 1 tvořená pull-up rezistorem a je otevřen tranzistor T3. Tranzistor T3, který drží celé zapojení aktivní do doby, než se pin mikrokontroléru přepne na výstupní, s log. 0. Poté je uzavřen tranzistor T3 a tranzistor T2.



Obr. 13 Zapojení obvodu pro ovládání

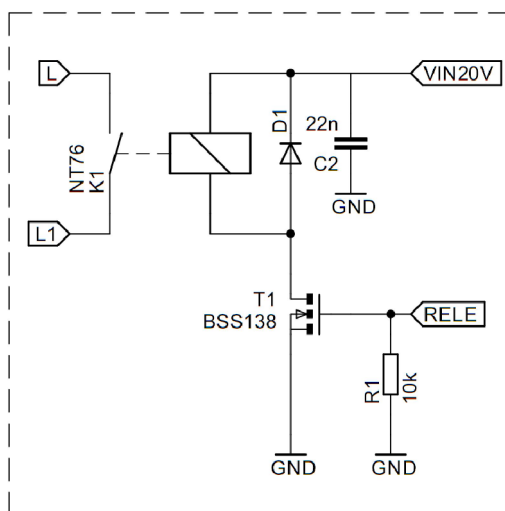
Jako další ovládací prvek je zapojeno nastavovací mikro-tlačítko připojené na pin MCLR/ GP3. Tento pin primárně slouží pro držení mikrokontroléru v resetu při programování nebo restartu. Zároveň má možnost být použit vstupní pin. Dioda je zde zapojena proto, aby při programování neproniklo vyšší programovací napětí cca 12 V= do napájecí větve obvodu s napětím 3,3 V=. Tento pin je doplněn 100kΩ pull-up rezistorem. [7]



Obr. 14 Zapojení vstupu pro nastavovací mikro-tlačítko

3.5 Spínání relé

Relé je spínáno FET tranzistorem s N kanálem, který má menší ztráty než bipolární tranzistor a také nezatěžuje tolik výstup mikrokontroléru. Odpor cívky relé je 2,8 kΩ. Pro sepnutí relé je nutný proud 10 mA. Paralelně k cívce relé je zapojen kondenzátor pro eliminaci poklesu napětí při sepnutí relé. Dioda byla antiparalelně zapojena. Při odpojení cívky relé od napájení dochází k přechodovému jevu. Dochází k uvolnění energie akumulované v cívce. To má za následek zvýšení napětí na cívce. Použitím diody se předchází poškození FET tranzistoru.

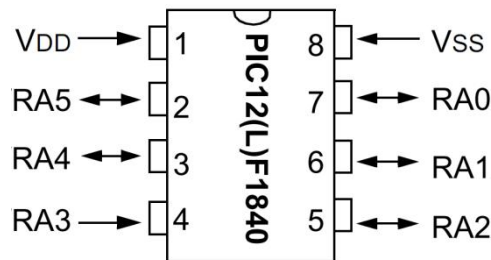


Obr. 15 Zapojení relé spínané pomocí MCU

3.6 Mikrokontrolér (MCU)

3.6.1 Volba vhodného kontroléru

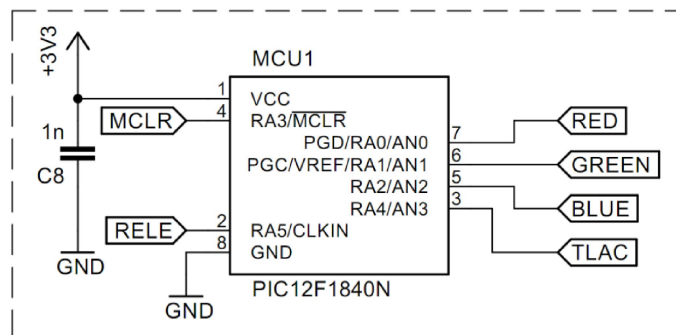
Pro tuto aplikaci byl vybrán mikrokontrolér firmy Microchip PIC 12F1840. Tento disponuje 4 kB pamětí programu, 256 Bytovou pamětí a 256 Bytovou EEPROM pamětí. Důvodem pro výběr mikrokontroléru byla také plocha, kterou zabere na DPS. Z důvodu velikosti bylo zvoleno provedení v pouzdře SO8, které lze snadno osadit na rozdíl od např. QFN. Cílem bylo maximálně využít mikrokontrolér, tak aby byly využity všechny piny.



Obr. 16 Rozložení pinů MCU PIC12F1840 [7]

3.6.2 Využití portů MCU

Na porty RA0, RA1, RA2 je připojena RGB LED v pořadí červená, zelená, modrá. Na pin RA3 bylo připojeno rozhraní MCLR¹ pro programování, a zároveň byl použit pro připojení mikro-tlačítka. Na pin RA4 je připojen kontakt z provozního tlačítka. Pin RA4 musí měnit svoje nastavení ze vstupního na výstupní podle potřeby. Na pin RA5 je připojen gate FET tranzistoru pro spínání relé. Jako zdroj hodinového signálu byl použit interní oscilátor mikrokontroléru nastavený na frekvenci 4 MHz. Piny RA0/DATA, RA1/CLOCK jsou zároveň používány při programování.

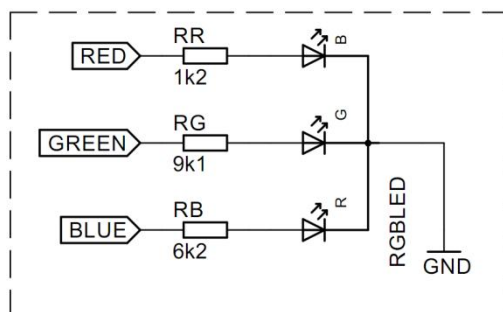


Obr. 17 Zapojení portů MCU

¹ MCLR – Externí reset používaný během programování MCU pomocí ICSP

3.7 RGB LED

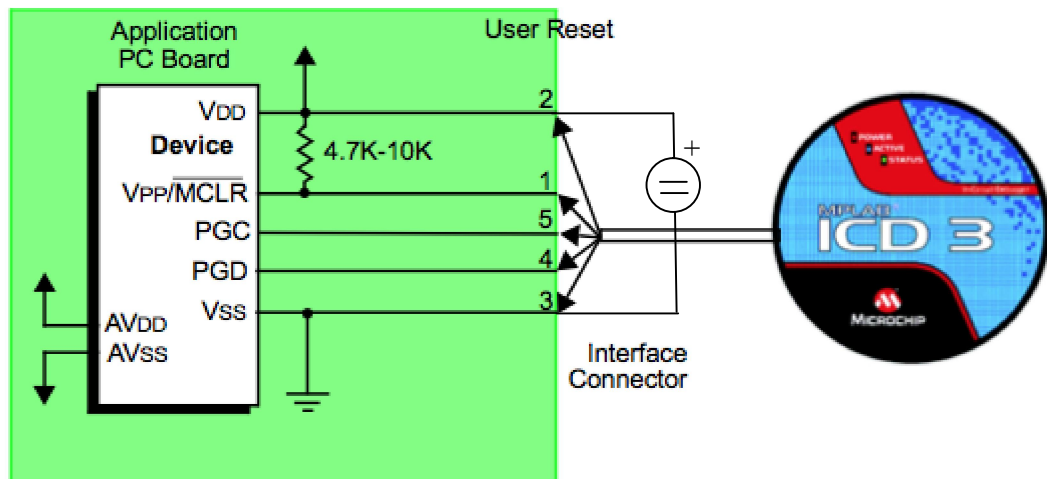
Pro signalizaci stavu zařízení byla využita RGB LED se společnou katodou. Pro rovnoměrnější jas RGB LED byly použity různé hodnoty předřadných rezistorů pro jednotlivé barvy. Nelze dosáhnout naprosto rovnoměrného rozložení z důvodu konstrukce difuzéru LED RGB. Výsledný vjem však může být velice subjektivní.



Obr. 18 Zapojení rezistorů a RGB LED

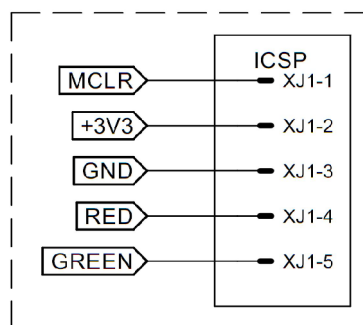
3.8 Programování pomocí ICSP

Tímto rozhraním probíhá programování zařízení. Pomocí tohoto rozhraní je možné program debuggovat – ladit přímo v aplikaci. Použitím programovacího rozhraní se usnadňuje i práce při odladování programu. Během programování je možné mikrokontrolér napájet přímo z programátoru, kde je možné nastavit napětí, v tomto případě 3,3 V=. Nicméně je vhodné při programování použít pro napájení externí zdroj 3,3 V=.



Obr. 19 Standardní zapojení programátoru pomocí ICSP (zdroj: Microchip, upraveno)

Při výrobě většího množství kusů, kdy se nepočítá se změnami programu, se nemusí osazovat konektor, protože stačí pouze zasunout do prokovených otvorů piny konektorů jako programovací hlavu. Toho se v tomto případě nevyužívá, protože se počítá s budoucím vylepšením programu. Je připojen plochý kabel s pěti-pinovou dutinkovou lištou s roztečí 2,54mm (0,1“). Rozteč otvorů na DPS je 1,27 mm (0,05“), stejně jako u plochého kabelu.



Obr. 20 Programovací konektor ICSP

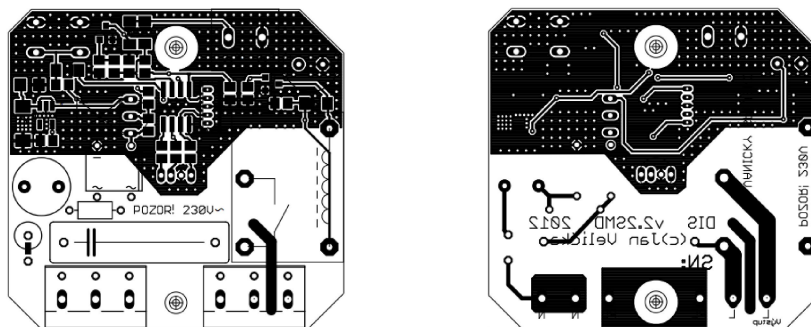
3.9 Deska plošných spojů

Tvar a rozměry DPS byly voleny s ohledem na umístění v elektroinstalační krabici. Rozměry DPS jsou 48×50 mm se zkosenými rohy. DPS je oboustranná (TOP, BOTTOM) s elektrolyticky prokovenými otvory. Z důvodu úspory prostoru je většina komponent v provedení SMD, osazované pouze z jedné strany, strany směrem do elektroinstalační krabice. Pro upevnění slouží otvory $\varnothing 3,2$ mm s roztečí 38 mm. Vzhledem k tomu, že silnoproudá a slaboproudá část nejsou galvanicky odděleny, jsou alespoň maximalizovány vzdálenosti těchto částí. Pro zvýšení elektrické pevnosti mezi spínacími kontakty relé byla odfrézována izolační mezera šířky 1,6 mm. Návrh DPS byl proveden s ohledem na doporučená návrhová pravidla pro DPS a minimalizaci vlivu elektromagnetického rušení.

- Materiál FR4 1,6 mm
- Nepájivá maska - zelená (Epoxy)
- Sítotiskový servisní potisk - bílý
- Pocínované kontakty

Pro návrh DPS bylo použito softwaru Eagle 6.2.0. Při návrhu byly vytvořeny nové knihovny součástek, případně upraveny původní. Výroba byla zadána firmě ITEADstudio z Hongkongu. U DPS bylo výrobcem provedeno testování elektrické správnosti.

Osazování DPS bylo prováděno do cínové pájecí pasty, která byla nanesena na plošky pomocí injekční stříkačky s kanylou pro pastozní materiály a gely (jehla bez zábrusu). Do této pasty byly vloženy SMD součástky pomocí pinzety. Po kontrole byly součástky pájeny horkovzdušnou pájecí stanicí. V případě, že některá ze součástek nebyla zcela na pájecích ploškách, bylo nutné provést korekci pomocí pinzety. Pájení THD součástek bylo provedeno pomocí mikropáječky a trubičkové pájky Fluitin 38Pb60Sn2Cu.

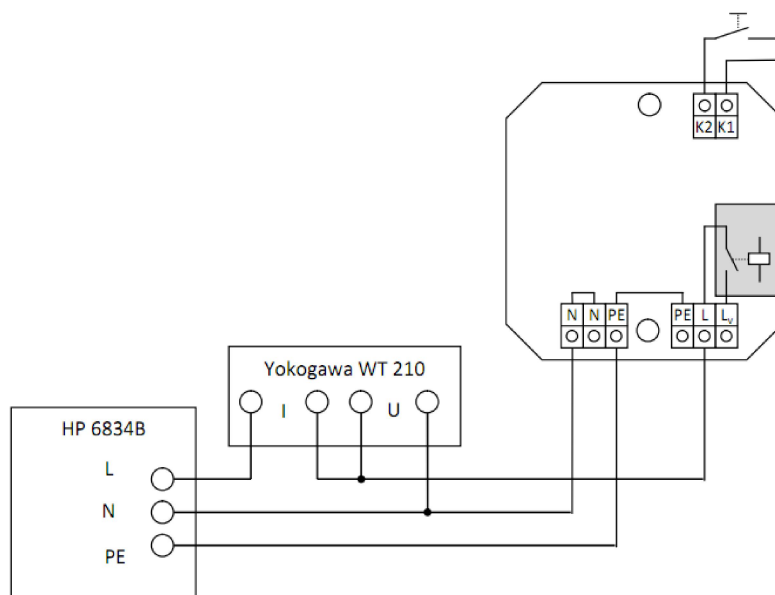


Obr. 21 Předloha DPS strana a) Horní (Top), b) Dolní (Botom)

4 Testování a měření hardwaru

4.1 Měření

Měřením byly zjišťovány parametry hlavně vztahující se ke spotřebě elektrické energie. Bylo zapotřebí změřit vlastnosti v rozsahu síťového napětí dle ČSN EN 50 160 a zjištění vlivu sepnutí relé na spotřebu zařízení. Dále byl zjišťován spodní limit napájecího napětí, kdy ještě dojde k sepnutí relé. K měření byla použita měřicí pracoviště v laboratořích EB419 a EB416, která jsou pro toto měření vybavena a zabezpečena před úrazem elektrickým proudem. Pro měření byl použit stolní wattmetr YOKOGAWA WT210 (s. č.: 27D517684) a programovatelný zdroj HP 6834B (s. č.:US360103).

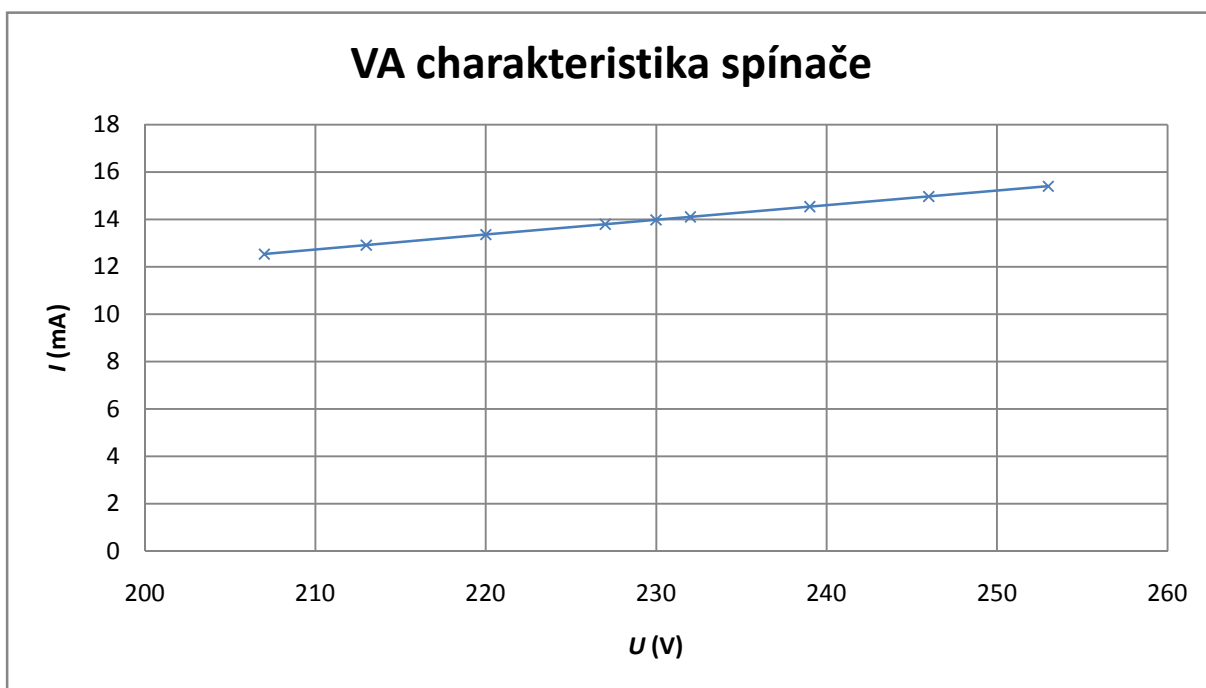


Obr. 22 Schéma zapojení pro měření

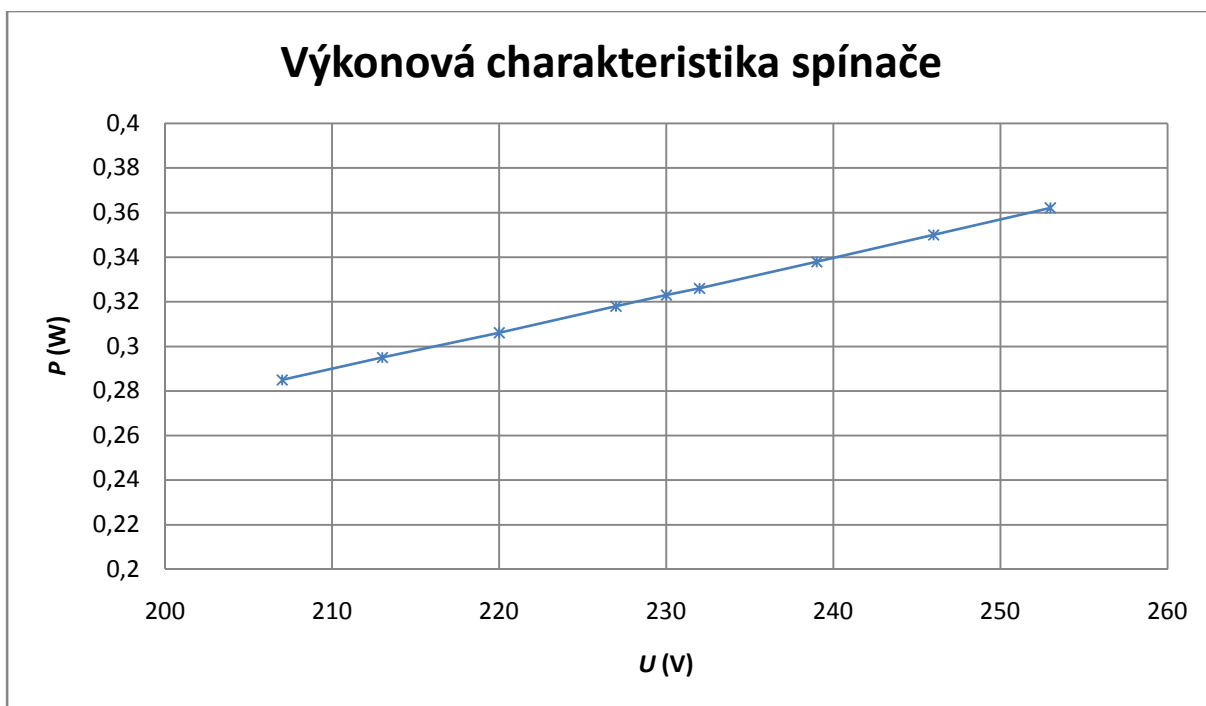
Měřením byly zjišťovány parametry: napětí, proud, výkon, jalový výkon, zdánlivý výkon, účinník, PF. Protože dle normy ČSN EN 50160 ed.3 (Parametry kvality elektrické energie) může být síťové napětí v rozsahu 207-253 V~, bylo nutné změřit parametry v tomto rozsahu.

Tab. 1 Tabulka naměřených hodnot

U (V)	I (mA)	P (W)	Q (var)	S (VA)	φ (°)	PF (-)
207	12,54	0,285	2,582	2,598	83,7	0,1098
213	12,92	0,295	2,737	2,754	83,9	0,1070
220	13,36	0,306	2,925	2,941	84,0	0,1041
227	13,79	0,318	3,116	3,132	84,2	0,1013
230	13,98	0,323	3,201	3,218	84,2	0,1003
232	14,10	0,326	3,257	3,273	84,3	0,0997
239	14,53	0,338	3,458	3,474	84,4	0,0974
246	14,97	0,350	3,667	3,683	84,6	0,0949
253	15,40	0,362	3,881	3,898	84,7	0,0929



Obr. 23 Graf závislosti proudu na napájecím napětí



Obr. 24 Graf výkonové charakteristiky

Vliv sepnutí relé na spotřebu byl zjišťován měřením při nominálním napětí (230 V~), jak je uvedeno v tabulce Tab. 2. Připojením impedance cívky relé dochází k poklesu spotřeby.

Tab. 2 Měření závislosti parametrů na stavu relé

Stav	U (V)	I (mA)	P (W)	Q (var)	S (VA)	PF (-)	φ (°)
Relé-zap	230	13,98	0,319	3,201	3,217	0,0991	84,3
Relé-vyp	230	13,98	0,322	3,201	3,218	0,1	84,3

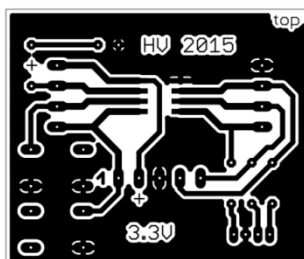
Byla také zjišťována dolní limitní hodnota napájecího napětí, při kterém zařízení ještě fungovalo správně, tedy došlo k sepnutí relé viz. Tab. 3.

Tab. 3 Parametry při dolním limitu napájecího napětí

U (V)	I (mA)	P (W)	Q (var)	S (VA)	φ (°)	PF (-)
150	9,2	0,2	1,37	1,39	81,6	0,1461

4.2 Testování

Pro vývoj a testování byla vyvinuta testovací DPS do nepájivého pole. Tak byly testovány vlastnosti MCU a RGB LED. Výrazně tak byl usnadněn vývoj softwaru z důvodu jednoduššího zapojení. Tato DPS je navržena jako jednostranná, s SMD mikrokontrolérem a ostatními součástkami v THD provedení. Pro zapojení do nepájivého pole slouží dvě precizní pinové lišty s roztečí 2,54 mm. Pro připojení programátoru je určena pěti-pinová lišta, zapojena dle ICSP.



Obr. 25 Předloha vývojové DPS



Obr. 26 Vývojové a testovací pracoviště

4.3 Opravy

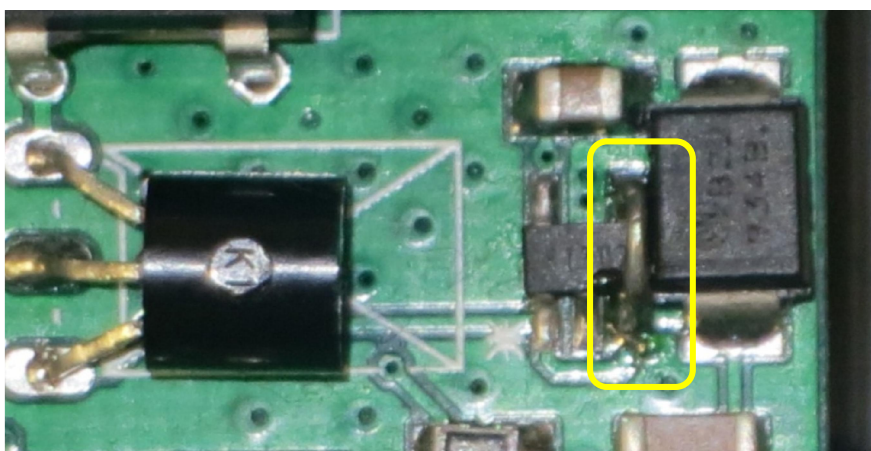
V průběhu řešení byly zjištěny následující nedostatky a závady, které byly odstraněny, jak je popsáno níže.

4.3.1 Snížení činných ztrát

Pro omezení činných ztrát byl odstraněn rezistor R9 v napájecím předřadníku. Tento rezistor byl nahrazen propojkou.

4.3.2 Úprava propojení LDO stabilizátoru

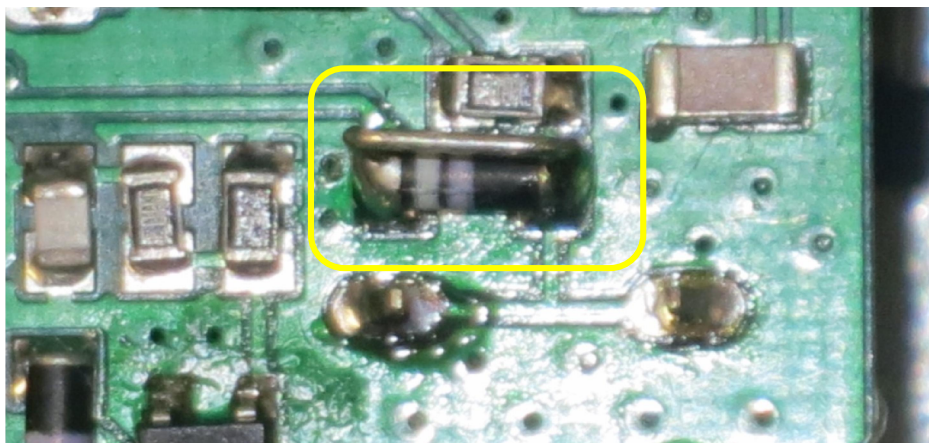
Z důvodu chyby (záměny) při definici součástky na DPS, bylo nutné propojit piny 3 a 4. Jedná se o piny „Shutdown“ a „Not connected“, z tohoto důvodu bylo možné provést opravu pouhou propojkou, jak je zvýrazněno na Obr. 27.



Obr. 27 Oprava zapojení LDO stabilizátoru

4.3.3 Zprovoznění nastavovacího mikro-tlačítka

Dle katalogového listu MCU (Příloha E), je uvedeno připojení vstupu MCLR přes diodu, z důvodu ochrany zbylých obvodů, před vyšším programovacím napětím. Úbytek napětí na této diodě ($0,65V \approx$), způsobil hodnotu napětí na vstupu MCU v zakázaném pásmu (cca $2,65V \approx$). Toto bylo odstraněno zkratováním diody (Obr. 28). Případně je možné diodu nahradit propojkou.



Obr. 28 Oprava zapojení vstupního obvodu tlačítka

5 Návrh softwaru

Software je navržen jako stavový automat s konečným počtem stavů. Jako nástroj pro popis funkce systému byly využity UML diagramy.

5.1 Popis funkcionality

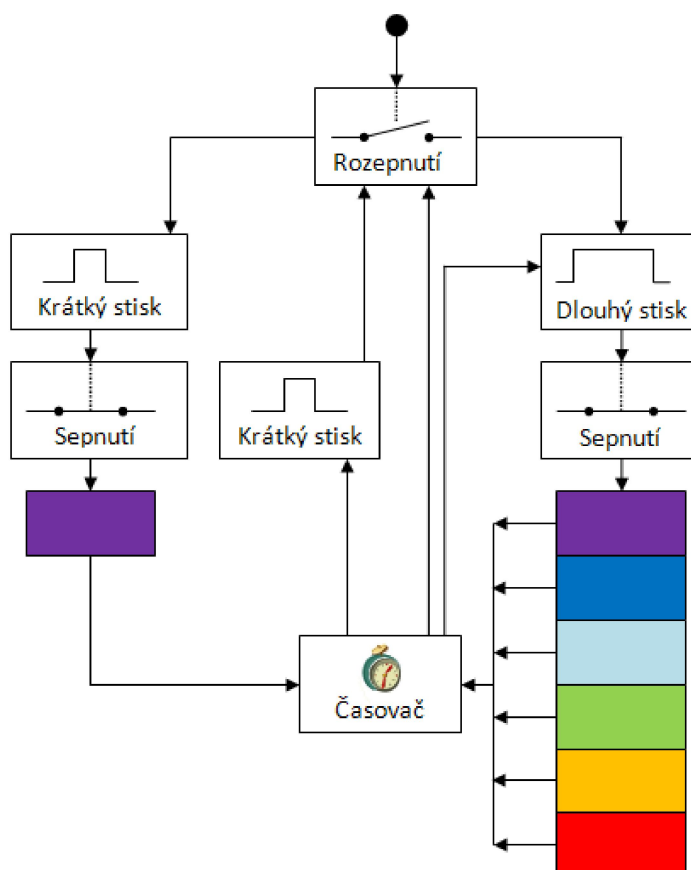
Zařízení čeká v rozepnutém stavu a dle délky stisku tlačítka je nastaven časovač. Pokud je následující stisk tlačítka kratší než je „Dlouhý stisk“², dojde k rozepnutí, tedy přechodu do výchozího stavu, jedná se tedy o „Krátký stisk“³. Pokud je stisk delší, než „Dlouhý stisk“, začnou se postupně měnit barvy signalizující volený čas. Po uvolnění stisku tlačítka dochází k postupnému odečítání času časovače. Jednotlivé intervaly jsou postupně signalizovány příslušnými barvami LED. Při uplynutí posledního časového intervalu (signalizováno fialovou barvou, resp. modrou barvou) dojde k rozepnutí, jak je možné vidět na Obr. 29 a Obr. 30. Je možné zvolit mezi třemi šesti-časovými a třemi dvou-časovými režimy. Tato volba se provádí stiskem nastavovacího mikro-tlačítka, kdy se stisky tohoto mikro-tlačítka přepínají režimy. Potvrzení se provádí stiskem tlačítka (hlavní uživatelské tlačítko) (Obr. 31).

² Dlouhý stisk – v tomto případě je hodnota nastavena na 1 s.

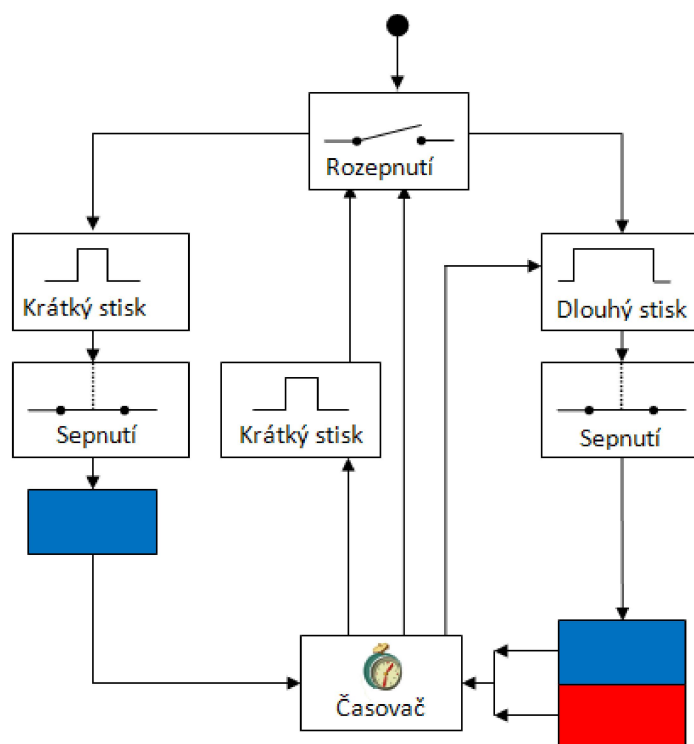
³ Krátký stisk – v tomto případě se jedná o stisk tlačítka na dobu menší než 1 s.

5.2 Koncepční návrh

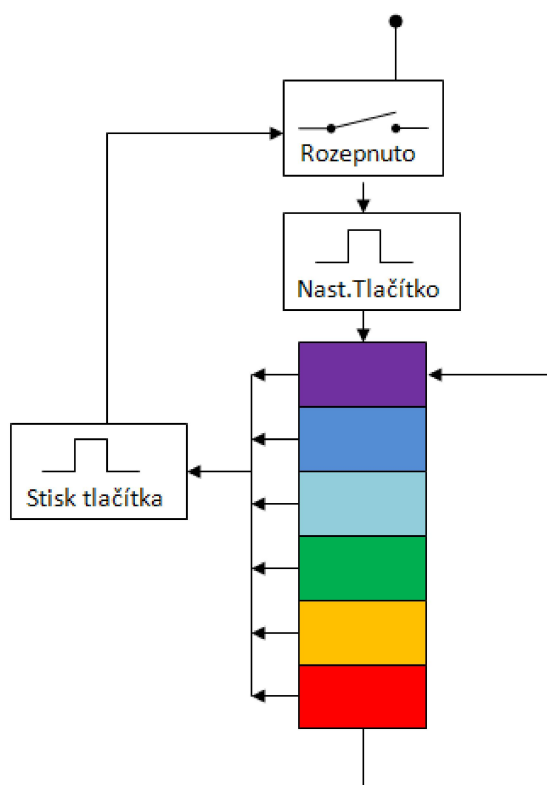
Bylo nutné definovat, jak se zařízení chová z vnějšího pohledu, z pohledu uživatele. Toto je popsáno v kapitole 5.1. Na základě toho bylo možné optimalizovat ovládání a následně i program. Bylo definováno, jak se zařízení zachová při určité události, případně jakou událost zařízení vyvolá. Graficky je znázorněna hlavně signalizace režimů pomocí barev a interakce se stisky tlačítek. Toho bylo využito při následném návrhu stavového automatu.



Obr. 29 Koncepční schéma šesti-časového režimu



Obr. 30 Konceptní schéma pro dvou-časový režim

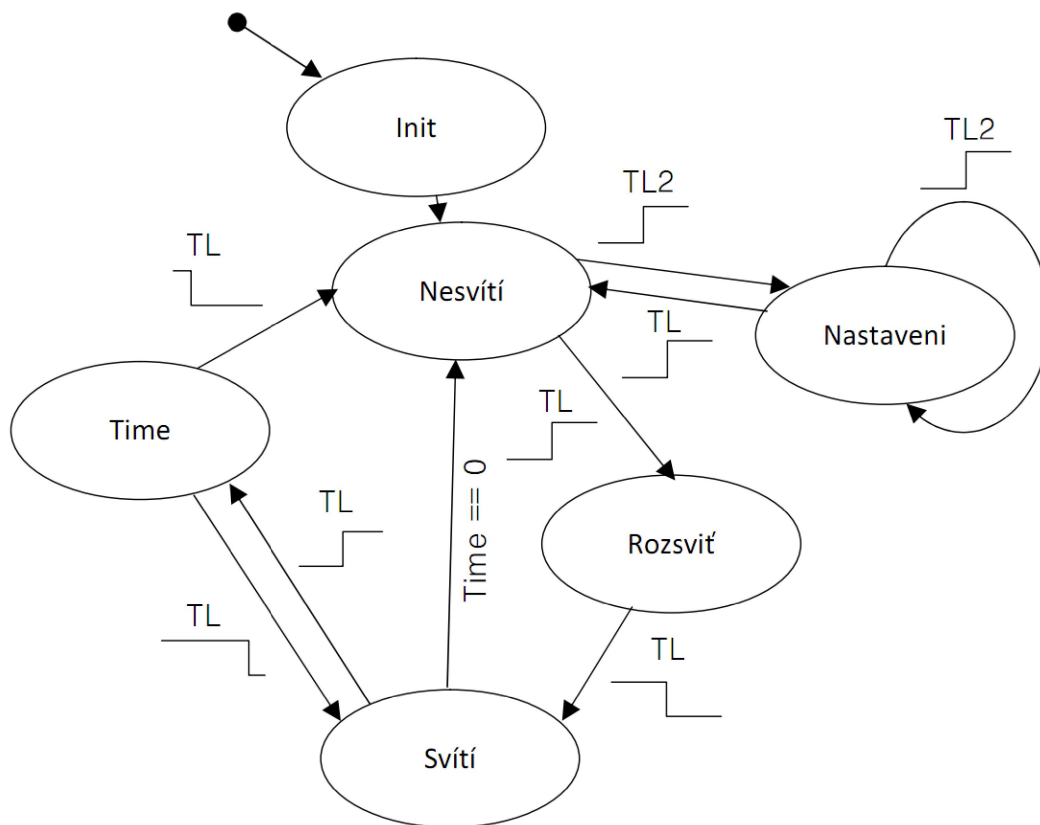


Obr. 31 Konceptní schéma pro nastavování režimu

5.3 Stavový diagram

Po nadefinování stavů a podmínek přechodů z koncepčního diagramu byl sestaven stavový automat (Obr. 32). V diagramu stavového automatu je možné vidět vazby mezi jednotlivými stavy. TL je kontakt hlavního uživatelského tlačítka, TL2 je kontakt nastavovacího mikro-tlačítka.

Ve výchozím stavu se zařízení nachází ve stavu „Nesvíť“. Z toho může přejít do stavu „Rozsviť“ nebo do stavu „Nastavení“. Při stisku tlačítka TL přejde do stavu „Rozsviť“, kdy na základě délky stisku dojde k předání parametru časovači a přechodu do stavu „Svíť“. V případě opětovného stisku tlačítka dochází k přechodu do stavu „Time“, kdy je vyhodnocována doba stisku tlačítka TL. V případě, že je stisk tlačítka TL kratší než zvolený „dlouhý stisk“, jedná se o „krátký stisk“ a přechod do stavu „Nesvíť“. Je-li délka stisku TL delší než zvolená délka, dochází ke znovu nastavení parametrů časovače a návrat do stavu „Svíť“. V případě uplynutí nastaveného času přechází do výchozího stavu „Nesvíť“. Stav „Init“ slouží pro načtení parametrů konfigurace zařízení. Ve stavu „Nastavení“ probíhá přepínání mezi režimy spínání pomocí stisků TL2.



Obr. 32 Diagram stavového automatu

6 Implementace do mikrokontroléru

Tato kapitola obsahuje popis funkcí použitých v MCU. Jako vývojové prostředí bylo použito MPLAB X 3.0 od firmy Microchip. Pro zápis do mikrokontroléru byl použit programátor a debugger ICD3 od firmy Microchip. Protože spolupráce mezi programátorem a vývojovým prostředím nebyla spolehlivá, byla používána aplikace MPLAB IPE 3.0, pomocí které byl prováděn zápis do mikrokonroléru.

Pro implementaci programu byl zvolen programovací jazyk C pro 8bitové mikrokontroléry Microchip. Oproti standardnímu ANSI C se liší jen minimálně. Je však rozšířen například o další datové typy (např.: `boolen`, `int8`), a tak je možné ušetřit paměť vhodnou volbou velikostí proměnné. Pro kompilaci byl použit kompilátor XC8 určený pro 8bitové mikrokontroléry Microchip. Tento kompilátor vytvoří soubor ve formátu `*.hex`, který je následně nahrán do mikrokontroléru. Kompletní zdrojové kódy programu jsou uvedeny v Příloze C.

6.1 Popis funkcí

Funkce – `main()`

Funkce `main()` je hlavní programová funkce, které se zavolá ihned po připojení napájení mikrokontroléru. V hlavní programové smyčce probíhá stavový automat, ve kterém se provádí ostatní funkce.

Přerušení - `interrupt isr ()`

Přerušení je vyvoláno přetečením registru čítače, toto je vyvoláno 400/s. Počítáním těchto přerušení může být dosaženo libovolného časového intervalu většího než 1/400 s. Pro tuto obsluhu se používá funkce `timer0()`. Pro další vyvolání přerušení je nutné vynulovat registry `GIE` a `TMR1IF`.

Funkce - `init()`

Pomocí této funkce je prováděna inicializace mikrokontroléru pro toto zařízení a jsou nastaveny parametry pro přerušení, vstupně-výstupní porty a další. Dále funkce provádí vypnutí nepotřebných komponent a načtení nastavení z paměti EEPROM a uložení do paměti SRAM.

Funkce - `timer0()`

Tato funkce obstarává výpočty a čítače související s časováním. Protože tato funkce je volána 400 krát za sekundu, je toto použito jako časová základna pro výpočet sekund, minut a hodin. Také v této funkci probíhá hlídání, zda je tlačítko stisknuto „krátce“ nebo „dlouze“, kdy je toho rozhodnuto podle hodnoty nastavené v `TL_LONG`.

Funkce - `nesvi()`

Tato funkce je volána ve stavu „`NESVITI`“, rozpojí relé a zhasne RGB LED. Zároveň čeká na stisk tlačítka nebo nastavovacího mikro-tlačítka. Dle stisknutého tlačítka dojde ke změně stavu. Je možné

nastavit barvu, která má svítit ve stavu „NESVITI“, jako orientační signalizace. V případě stisku nastavovacího mikro-tlačítka ke změně stavu na „SERVIS“.

Funkce - rozsvit()

Funkce zajišťuje obsluhu ve stavu „ROZSVIT“, který nastane ve chvíli, kdy sepne tlačítko. Následně se sepne kontakt relé a zavoláním funkce `time()` se zjišťuje na jakou hodnotu je nastaven čas dle doby stisku tlačítka.

Funkce - sviti()

Tato funkce je volána ve stavu „SVITI“. V případě stisku tlačítka změni stav na „TIME“. Dále je z této funkce volána funkce `watch()`, která hlídá čas zbývajících v časovači.

Funkce - time()

Tato funkce je používá ve dvou stavech „ROZSVIT“ a „TIME“. Slouží ke zjištění doby, po kterou je stisknuto tlačítko. Ve stavu „TIME“ navíc rozlišuje mezi krátkým a dlouhým stiskem tlačítka. Při krátkém stisku dojde k přechodu do stavu „NESVITI“

Funkce - servis()

Funkce je volána ve stavu „SERVIS“. Při stisku nastavovacího mikro-tlačítka se postupně přepínají přednastavené sady časů. Potvrzení se provádí stiskem hlavního tlačítka, zvolené nastavení se ukládá do EEPROM. Parametry pro časovač jsou uloženy do paměti SRAM. Signalizace servisního režimu je změnou frekvence a střídou blikání RGB LED.

Funkce - blik()

Funkce se používá pro blikání RGB LED, kdy je možné přepínat mezi dvěma zvolenými barvami, měnit frekvenci, střidu a povolovací bit. Je volána v každém přerušení od časovače `tmr0`. Parametry se vypočítají zavoláním funkce `vypocet()`. Vzhledem k tomu, že parametry jsou odvozené od počtu přerušení za sekundu, je toto vhodné pro viditelné frekvence blikání, protože není možné zajistit jemnost dělení střídou. Maximální frekvence blikání je v tomto případě 200 Hz při 50 % střídě.

Funkce - color()

Funkce slouží ke změně barvy RGB LED. Vstupním parametrem je číslo barvy a tím je usnadněna práce s barvami RGB LED.

- 0 Žádná
- 1 Fialová
- 2 Modrá
- 3 Tyrkysová
- 4 Zelená
- 5 Žlutá
- 6 Červená
- 7 Bílá

Funkce - `doba_stisku()`

Tato funkce rozlišuje, zda je stisk tlačítka krátký nebo dlouhý. K tomu je využívána hodnota získána od přerušení časovače, a když překročí hodnotu `TL_LONG`, jedná se o dlouhý stisk tlačítka.

Funkce - `watch()`

Tato funkce je volána ve stavu „SVITII“ a kontroluje zbývajících čas časovače. V případě, že jsou hodnoty v časovači nižší, než je čas v uložený v předchozím prvku pole, dojde ke snížení indexu pole a změny barvy RGB LED. Když zbývá v posledním intervalu poslední půlminuta, dojde pomocí změny frekvence a střidy k upozornění na zbývajících běh poslední půlminuty posledního intervalu. Když je čas časovače roven nule, je změněn stav na „NESVITII“.

Funkce - `vypocet()`

Funkce slouží pro výpočet hodnot parametrů blikání RGB LED. Těchto parametrů využívá funkce `blik()`. Z hodnot zadané frekvence (Hz) a střidy (%) je dopočítáno, kolik přerušení má být v prvním stavu barvy a kolik ve druhém stavu barvy.

Příklad výpočtu pro $f = 4$ Hz a střidu = 60 %:

$$n_{\text{přerušení}} = 400 \quad (1)$$

$$f_{\text{přerušení}} = n_{\text{přerušení}}(s^{-1}) = 400 \text{ Hz} \quad (2)$$

$$n_{\text{přerušení_perioda}} = \frac{f_{\text{přerušení}}}{f} = \frac{400}{4} = 100 \quad (3)$$

$$n_{\text{přerušení_B1}} = \frac{n_{\text{přerušení_perioda}} \cdot \text{střida}}{100} = \frac{100 \cdot 60}{100} = 60 \quad (4)$$

$$n_{\text{přerušení_B2}} = \frac{n_{\text{přerušení}}}{\text{perioda}} - n_{\text{přerušení_B1}} = 100 - 60 = 40 \quad (5)$$

Kde:

- $n_{\text{přerušení}}$ - hodnota charakterizující počet přerušení za jednu sekundu.
- $n_{\text{přerušení_perioda}}$ - hodnota určující počet přerušení na jednu periodu bliknutí.
- $n_{\text{přerušení_B1}}$ - hodnota určující počet přerušení, kdy je rozsvícena první barva.
- $n_{\text{přerušení_B2}}$ - hodnota určující počet přerušení, kdy je rozsvícena druhá barva

Funkce – `write_EE()`

Funkce slouží pro zápis do paměti typu EEPROM. Funkce obsahuje nutnou sekvenci předepsanou v katalogovém listu [7]. Jako parametry funkce se předávají adresa v paměti EEPROM a data (byte).

Funkce – read_EE()

Funkce slouží pro vyčtení z paměti EEPROM. Jako parametr funkce je zadávána adresa v paměti EEPROM. Návrátová hodnota funkce je obsah na adrese v paměti EEPROM typu byte.

Funkce – rezimy_led()

Tato funkce je používána v případě nutnosti měnit barvu RGB LED podle počtu časů. Pro režimy kdy jsou využívány pouze dva kroky, je nutné mít barvy v pořadí, které je nastaveno (např. modrá, červená). V případě použití šesti-časového režimu jsou barvy v pořadí (fialová, modrá, tyrkysová, zelená, žlutá, červená).

Paměť EEPROM

Do paměti EEPROM jsou ukládány aktuální nastavení, výběr ze tří šesti-časových sad a tří dvou-časových sad. Dále je možné zápisem do paměti změnit barvu signalizující výchozí stav. Zápis do této paměti je možný programátorem ICD3.

7 Testování celého zařízení

Testování bylo prováděno jak v laboratorních, tak provozních podmínkách. Pomocí mobilních prototypů i pevných instalací. Provozními podmínkami je myšlena domácí elektroinstalace, pro kterou je zařízení určeno.



Obr. 33 Ukázka kompletace zařízení

Pro demonstrační účely byl vytvořen modul tlačítka a zásuvky. Na tomto modulu bylo prováděno testování z hlediska uživatelské přívětivosti. Protože jinak je ovládáno zařízení pomocí mikrospínače jak je tomu na vývojové DPS nebo tlačítkem ABB (řada Tango). Z toho vyplývají i jiné vlastnosti při sepnutí z hlediska odskoku kontaktu a ošetření tohoto jevu. Výhodou byla možnost snadno změnit program a zároveň mobilita pro demonstraci.



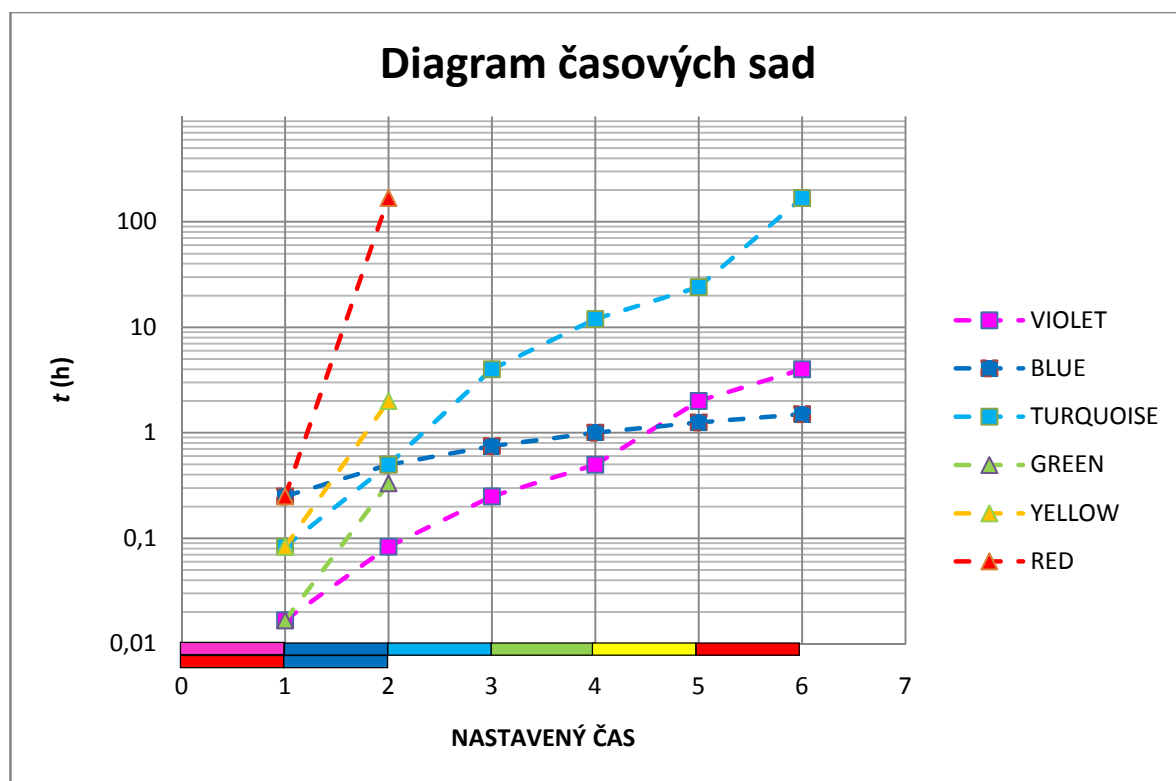
Obr. 34 Demonstrační modul konfigurovatelného spínače

Během testování se ukázalo, že je vhodnější, aby byl prodloužen čas pro „dlouhý“ stisk tlačítka na 1 s. Taktéž se ukázalo, že i dobu mezi přepnutími barev je zapotřebí prodloužit na 0,5 s. Protože z hlediska uživatelského komfortu je nevhodné, aby docházelo k chybnému nastavení.

V rámci testování byly upravovány sady časů (Tab. 4), aby bylo možné sady používat pro různé aplikace. Jedná se především o rozdílné potřeby času pro svícení a větrání. Proto byly tyto sady voleny, co nejuniverzálněji s ohledem na návaznost jednotlivých hodnot.

Tab. 4 Hodnoty časových sad - hodnoty v paměti MCU

Sada 6t	t	1	2	3	4	5	6
	hodiny	0	0	0	0	2	4
	minuty	1	5	15	30	0	0
	hodiny	0	0	0	0	0	0
	minuty	15	30	45	60	75	90
	hodiny	0	0	4	12	24	168
	minuty	5	30	0	0	0	0
Sada 2t	t	1	2				
	hodiny	0	0				
	minuty	1	20				
	hodiny	0	2				
	minuty	5	0				
	hodiny	0	168				
	minuty	15	0				



Obr. 35 Diagram časových sad

8 Závěr

V práci byl zpracován přehled smart-home technologií, které je možné použít jako doplněk do stávající elektroinstalace. Jde tedy o bezdrátová řešení, která jsou vhodná pro rodinné domy a byty. Jednalo se o systémy firem Loxone a iNels. Dle zjištění je zajímavější systém iNels, který nabízí široké spektrum různých snímačů, akčních členů a jiných zařízení. Systém Loxone bývá použit tam, kde je kooperace s KNX.

V práci byly také hodnoceny, v rámci průzkumu trhu, výrobky s podobnou funkcí, jako má zařízení zpracovávané v této práci – Konfigurovatelný časový spínač. Jednalo se převážně o schodišťové automaty, případně zařízení prodlužující dobu svícení. Dle tohoto porovnání pravděpodobně není dnes na trhu zařízení, které by bylo schopné plně konkurovat zařízení navrženému a realizovanému v rámci této práce.

Byl navržen a realizován hardware pro konfigurovatelný časový spínač pro umístění do elektroinstalační krabice KU68. Hardware byl realizován oboustrannou DPS a kombinací SMD a THD součástek. Napájení bylo realizováno pomocí kapacitního předřadníku, usměrňovače a LDO stabilizátoru. Řízení zajišťuje 8bitový mikrokontrolér Microchip PIC 12F1840, který obstarává řízení relé na základě stisku tlačítka nebo časovače.

Měřením bylo zjištěno, že výkonová ztráta zařízení je 0,3 W, celkově má zařízení kapacitní charakter. Toto lze považovat za dostatečně nízké ztráty. Bohužel nebylo úspěšně realizováno odpojení od zdroje, čímž by bylo dosaženo v klidovém stavu nulové vlastní spotřeby.

Pro testování byla vytvořena testovací DPS do nepájivého pole, na které byl převážně testován software. Testováním byly zjištěny nedostatky, které byly úspěšně opraveny. Jednalo se o chyby v návrhu, které bylo možné odstranit propojkou, nebo přemostěním.

Návrh softwaru, který byl implementován do mikrokontroléru, probíhal v souladu s principy softwarového inženýrství. Na základě popisu funkcionality, kterou má zařízení mít, byly vytvořeny UML diagramy. Použité ULM diagramy usnadnily a zrychlily následnou implementaci. Byla vytvořena koncepční schémata pro lepší názornost fungování zařízení a navržen stavový automat, ve kterém byly definovány stavy a podmínky přechodů mezi stavy.

Implementace softwaru do mikrokontroléru byla realizována v jazyce C pro 8bitové mikrokontroléry. Byl implementován stavový automat. Pro měření času bylo použito přerušení od vnitřního časovače. Pro ukládání některých parametrů byla použita paměť EEPROM.

Během testování při použití dle určení – osazené v elektroinstalaci byla zjištěna nutnost upravit některá nastavení pro vyšší uživatelský komfort. Jednalo se především o některé časové prodlevy při stisku tlačítka.

Zařízení vytvořené v rámci této práce je provozuschopné a použitelné všude tam, kde je zapotřebí sepnout zátěž na určitou dobu. Jedná se především o spínání světel nebo ventilátorů. Velkou výhodou tohoto zařízení je široký rozsah spínaných časů (1 min – 255 hod) a možnost volby nejvhodnější šestice, resp. dvojice časů pro danou aplikaci.

Jako možné rozšíření lze uvést několik dalších režimů, například cyklické spínání a pseudo-náhodné spínání nebo jedno-časový režim. Případně je možné doplnění konfigurace softwaru prostřednictvím sériové linky. Následně se nabízí možnost týdenního plánování spínání, kdy nastavení bude prováděno pomocí aplikace v počítači. Dále by bylo možné rozšířit zařízení o bezdrátový komunikační modul, například ESP8266 s wifi a sériovým rozhraním. Tím by se tento konfigurovatelný spínač dostal do kategorie zařízení „Internet of Things“.

Literatura

1. *Katalogový list CRM-4*. [online] Holešov : ELKOep.
2. *Katalogový list CRM-42*. [online] Holešov : ELKOep.
3. Balza, J. "Schodišťový" časový spínač. *Praktická elektronika*. 1999.
4. Loxone. LOXONE ŘEŠENÍ. [Online] 15. 3 2016. <http://www.loxone.com/cscz/chytry-dum/prehled.html>.
5. *Instalační manual Dům pod palem*. [online] Holešov : ELKOep, iNels.
6. *MCP1804*. [online] USA : Microchip Technology Inc., 2009.
7. *PIC12(L)F1840 Datasheet*. [online] USA : Microchip Technology Inc., 2011. 978-1-61341-183-4.
8. Abdulkadhim, H A a Hasan, T F. *DESIGN OF A SOFTWARE SYSTEM FOR FINITE STATE*. Diyala : Diyala Journal of Enigeering and Science, 2011. 1999-8716.
9. *Přístroj 3585 s krytem a rámečkem*. [online] : ABB.
10. Herout, Pavel. *Učebnice jazyka C - 1.díl*. : KOPP, 2005.

Seznam příloh

Příloha A: Schéma zapojení – 1 strana A4

Příloha B: Výkres umístění do KU68 – 1 strana A4

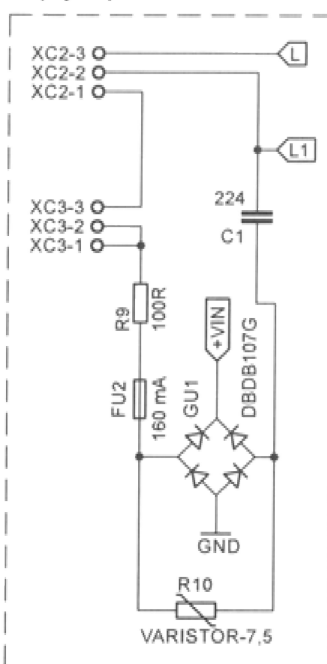
Příloha C: Zdrojový kód softwaru – 20 stran

Příloha D: Návod k použití konfigurovatelného spínače – 4 strany

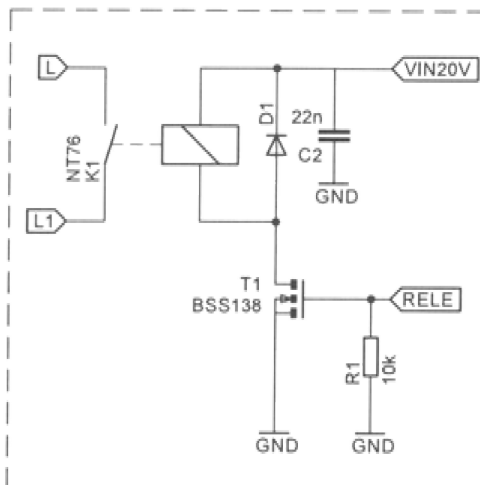
Příloha E: Katalogový list MCU PIC12F1840 - Příloha na CD – 382 stran

Příloha F: Katalogový list LDO Stabilizátoru MCP1804 - Příloha na CD – 38 stran

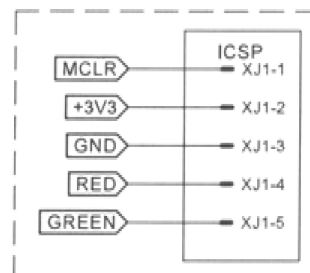
Napájecí předřadník



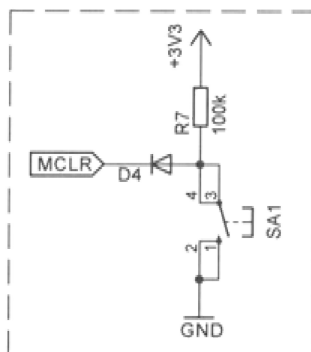
Spínání relé



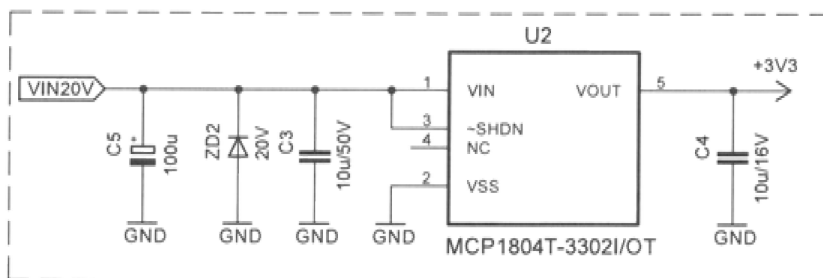
konektor ICSP



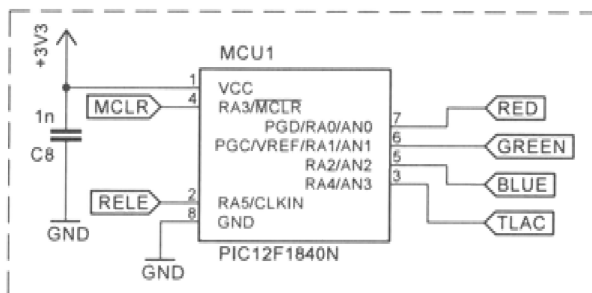
Ovládání



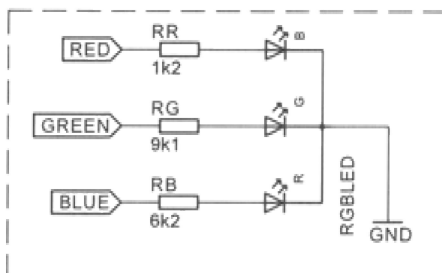
Stabilizace napětí



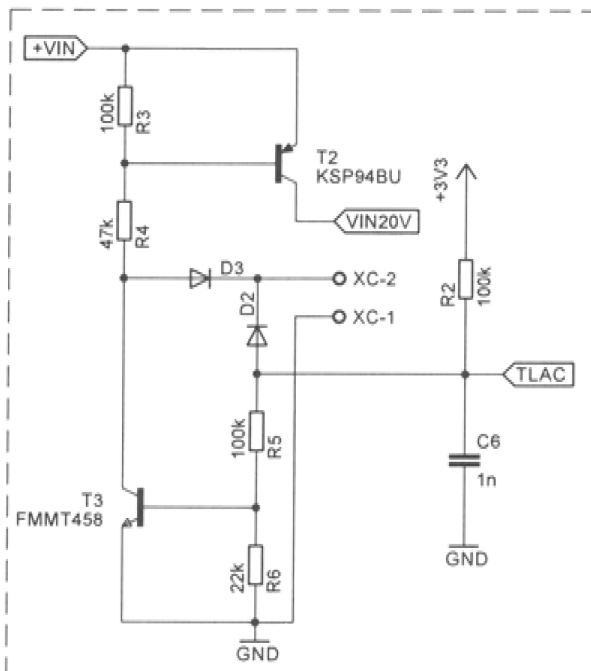
Mikrokontrolér



RGB LED



Ovládání



TITLE:

Příloha A
Konfigurovatelný spínač

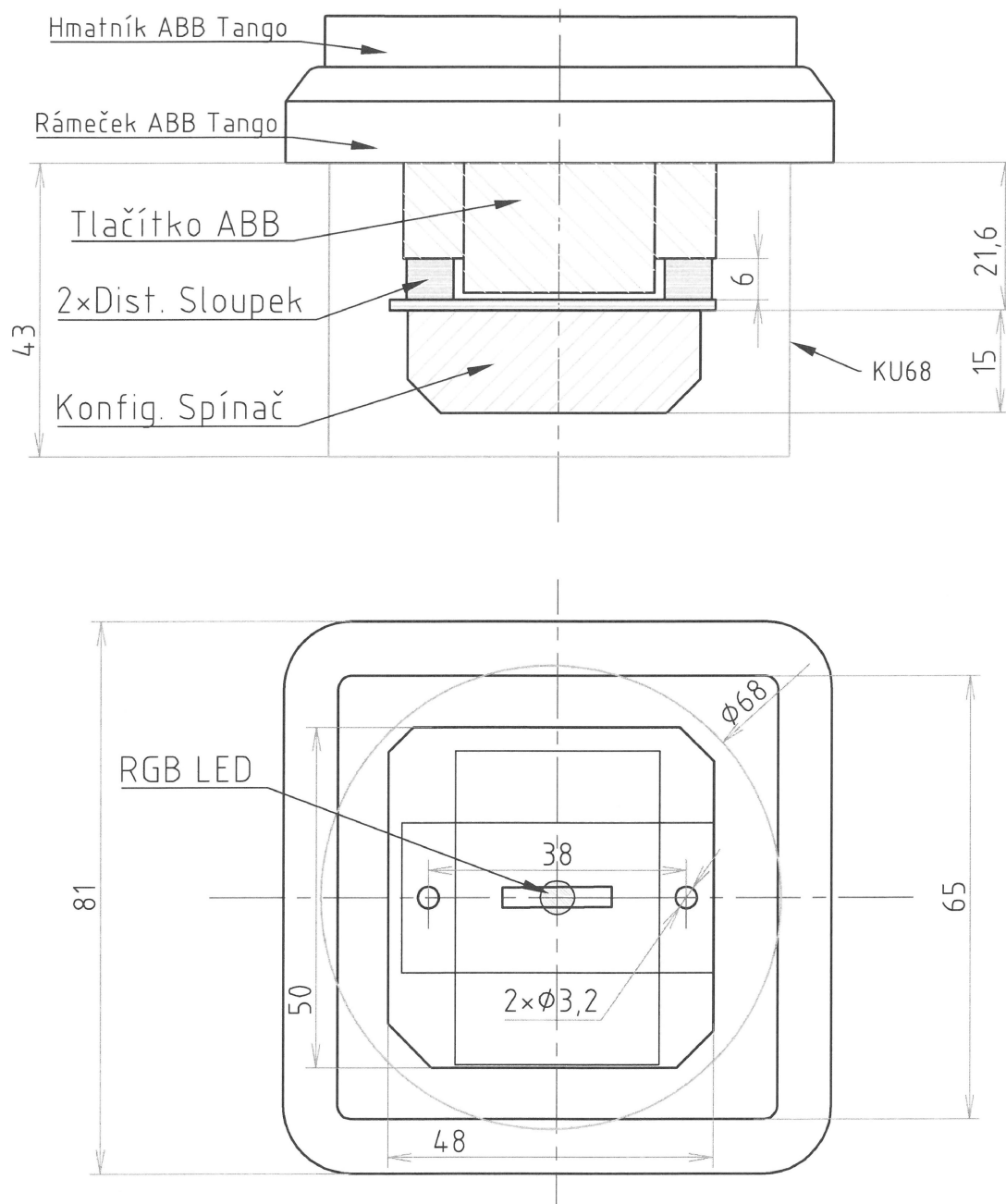
Jan Velička

Project: DIS_v2_2

Save: 6. 3. 2016 21:12:01

Sheet: 1/1

A4



				Přesnost		Materiál	Polotovár
				Tolerování		Čistá hmotnost	Hrubá hmotnost
				Měřítko	Tř. odp.	Promítání	Č. sestavy
Index	Popis změny	Dne	Podpisy	1:1			Č. seznamu položek
VŠB-TU Ostrava, FEI Katedra kybernetiky a biomedicínského inženýrství KAT450		Kreslil JAN VELIČKA Technický referent		Změna	Jazyk CZ		Č. dokumentu VSB-VEL0069-2015-02
Soubor PrilohaB.pdf		Přezkoušel		Datum vydání 15.2.2016		Název, doplňující název Příloha B Umístění v KU68	
				Formát	List		
				A4	1/1		

Příloha C: Zdrojový kód softwaru

Zdrojový kód - main.c

```
////////////////////////////////////
// (c)Jan Velicka
// 1/2015 - 2/2016
////////////////////////////////////

#include <xc.h>
#include "main.h"
#include <stdbool.h>
#include <stdint.h>

////////////////////////////////////
//VARIABLES
////////////////////////////////////

typedef enum{
    STAV_INIT,
    STAV_NESVITI,
    STAV_ROZSVIT,
    STAV_SVITI,
    STAV_TIME,
    STAV_UDRZBA
} TYP_STAV;

TYP_STAV stav;

volatile int16_t pulse;
volatile int16_t period;
volatile int8_t sec_icolor = 0;

const unsigned int8_t volby_h[3][7] = {
{0,0,0,0,0,2,4},
{0,0,0,0,0,0,0},
{0,0,0,4,12,24,168}};
const unsigned int8_t volby_m[3][7] = {
{0,1,5,15,30,0,0},
{0,15,30,45,60,75,90},
{0,5,30,0,0,0,0}
};
```

```

const unsigned int8_t volby_kd_h[3][3] = {
{0,0,0},
{0,0,2},
{0,0,168}
};
const unsigned int8_t volby_kd_m[3][3] = {
{0,1,20},
{0,5,0},
{0,15,0,}
};

volatile unsigned uint8_t casy_h[7] = {0,0,0,0,0,1,5}; //pole ve
kterem budou ulozene casy(v hodinach) (vychozí)
volatile unsigned uint8_t casy_m[7] = {0,1,2,5,15,0,0}; //pole ve
kterem budou ulozene casy(v minutach) (výchozí)

volatile bool TL_flag;
volatile bool TL_long;
volatile bool TL_short;
volatile bool blik_en;
volatile bool sRED;
volatile bool sGREEN;
volatile bool sBLUE;

volatile unsigned int count = 0; // pomocna promena pro tmr0
volatile unsigned int count_pom = 0; // pomocna promena pro tmr0
volatile unsigned int8_t count_minute = 0;
volatile unsigned int16_t count_color = 0;
volatile unsigned int8_t index = 0;
volatile unsigned int8_t c_index = 0;
volatile unsigned int8_t frekvence = 0;
volatile unsigned int8_t count_sec = 0;
volatile unsigned int8_t count_hour = 0;
volatile unsigned int count_TL = 0;
volatile unsigned int count_blik = 0;

const    int16_t pulse_sec = PULSE_PER_SEC ;
volatile unsigned int8_t nastaveni = 0;
volatile unsigned int8_t off_color = 0;
volatile unsigned int8_t pocet_stupnu = 0;

```

```

/////////////////////////////////////////////////////////////////
//FUNCTION PROTOTYPES
/////////////////////////////////////////////////////////////////
void init(void);
void timer0(void);
void nesvi(void);
void rozsvit(void);
void sviti(void);
void time(void);
void servis(void);
void blik(int bperiod,int bpulse ,int a,int b,bool en);
void color(int color_index);
void doba_stisku(void);
void watch(void);
void vypocet(int freq, int percent );
void write_EE(int8_t eadr, int8_t edata);
int8_t read_EE(int8_t eadr);
void rezimy_led(void);

/////////////////////////////////////////////////////////////////
//INTERRUPTS
/////////////////////////////////////////////////////////////////
void interrupt isr ()
{
    GIE = FALSE;

    /*****TIMER 0 *****/
    if(TMR0IF && TMR0IE)//kontrola stavu periferii
    {
        timer0();

        TMR0IF = FALSE; //nulovani priznaku preruseni od TMR0
    }
    /*****TRM0*****/

    /*****/
    GIE = TRUE;

}
/*****END INTERRUPTS*****/

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//MAIN
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void main()
{
    init();

    while(1)
    {
        switch(stav)
        {
            case STAV_INIT:
                init();
                break;

            case STAV_NESVITI:
                nesvi();
                break;

            case STAV_ROZSVIT:
                rozsvit();
                break;

            case STAV_SVITI:
                sviti();
                break;

            case STAV_TIME:
                time();
                break;

            case STAV_UDRZBA:
                servis();
                break;

            default :
                init();
        }
    }
    return;
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//END MAIN
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

```

////////////////////////////////////
//FUNCTIONS
////////////////////////////////////
/*****INIT*****/
void init()
{
    //////////////////////////////////
    // CONFIG
    //////////////////////////////////

    FVREN = OFF; // vypnuti napetove reference
    ANSELA = OFF;
    CPSON = OFF; // vypnuti kapacitnich vstupu
    SSPEN = OFF; // vypnuti serial
    CCP1CON = OFF;
    CM1CON0 = 0b00000111; // vypnuti ADC
    OPTION_REG = 0b00001010; // weak-up R, fosc/4, 1:8
    C1ON = OFF; // vypnuti komparatoru
    LATA = OFF; // vynulavani latch registru


    LATA4 = OFF;
    TRISA0 = OUTPUT;
    TRISA1 = OUTPUT;
    TRISA2 = OUTPUT;
    TRISA5 = OUTPUT;


    TRISA3 = INPUT;
    TRISA4 = INPUT;


    WPUA4 = ON;
    WPUA3 = ON;


    TMR0IE = 1; // povoleni preruseni TMR
    TMR0 = TMR0_REFILL; // nastaveni hodnoty pro TMR0
    PSA = 0; // prediv.
    PS0 = 0;
    PS1 = 0;
    PS2 = 0;


    TMR1IE = ON; // povoleni preruseni od TMR1
    TMR1 = TMR1_REFILL;
    TMR1CS0 = 0;
    TMR1CS1 = 0;

```

```

TMR1ON = 1;
TMR1GE = 0;

T1CKPS0 = 0;
T1CKPS1 = 0;

TLAC_DIR = INPUT;
stav = STAV_NESVITI;

RED = OFF;
GREEN = OFF;
BLUE = OFF;
RELE = OFF;

vypocet(FREQ_BLIK_STD, DUTY_BLIK_STD);
GIE = ON; //povoleni globalniho preruseni

nastaveni = read_EE(EESET);
off_color = read_EE(OFF_COLOR_ADR);

if(off_color >= 8) //kdyz bude FF, barva v nesviti
{
    off_color = 0;
}

if(nastaveni >= POCET_NASTAVENI) //kdyz bude FF
{
    nastaveni = 0;
}
if(nastaveni <= 2) //nacteni dle eeprom
{
    casy_h[0] = volby_h[nastaveni][0];
    casy_h[1] = volby_h[nastaveni][1];
    casy_h[2] = volby_h[nastaveni][2];
    casy_h[3] = volby_h[nastaveni][3];
    casy_h[4] = volby_h[nastaveni][4];
    casy_h[5] = volby_h[nastaveni][5];
    casy_h[6] = volby_h[nastaveni][6];

    casy_m[0] = volby_m[nastaveni][0];
    casy_m[1] = volby_m[nastaveni][1];
    casy_m[2] = volby_m[nastaveni][2];
    casy_m[3] = volby_m[nastaveni][3];
    casy_m[4] = volby_m[nastaveni][4];

```

```

        casy_m[5] = volby_m[nastaveni][5];
        casy_m[6] = volby_m[nastaveni][6];
        pocet_stupnu = 6;
    }
    if(nastaveni > 2 && nastaveni <= POCET_NASTAVENI-1)
//nacteni dle eeprom
    {
        casy_h[0] = volby_kd_h[nastaveni-3][0];
        casy_h[1] = volby_kd_h[nastaveni-3][1];
        casy_h[2] = volby_kd_h[nastaveni-3][2];
        casy_h[3] = 0;
        casy_h[4] = 0;
        casy_h[5] = 0;
        casy_h[6] = 0;

        casy_m[0] = volby_kd_m[nastaveni-3][0];
        casy_m[1] = volby_kd_m[nastaveni-3][1];
        casy_m[2] = volby_kd_m[nastaveni-3][2];
        casy_m[3] = 0;
        casy_m[4] = 0;
        casy_m[5] = 0;
        casy_m[6] = 0;
        pocet_stupnu = 2;
    }

}
/*****end int *****/

/*****NESVITI*****/
void nesvi()
{
    if(TLAC_DIR == 1)
    {
        RELE = OFF;
    }
    index = 0;
    //NIC();
    c_index = off_color;
    color(c_index);
    blik_en = FALSE;
    index = 0;
    sec_icolor = 0;

```

```

if( TLAC == CLOSE && TLAC_DIR == INPUT)//skok do rozsvit
{
    vypocet(FREQ_BLIK_SET,DUTY_BLIK_SET);
    stav = STAV_ROZSVIT;

}

if(NASTAV_TLAC == CLOSE)// skok do rezimu servis
{
    stav = STAV_UDRZBA;
}

}

/*****
/*****funkce timer0 *****/
void timer0()
{
    TMRO = TMRO_REFILL;    //nastaveni hodnoty pro TMRO
    count_pom++;

    if(count_pom >= pulse_sec-2 )
    {
        count_pom = 0;
        count_sec--;
    }

    blik(period , pulse,c_index,sec_icolor,blik_en);    //Blikání

    if(TLAC == CLOSE) // doba stisku TLAC
    {
        count_TL++;

        if(count_TL >= 1000) //jen osetreni pretečení
        {
            --count_TL;
        }
    }
    else
    {
        count_TL = 0;
    }

    if(count_sec <= 0) //odecitani z nastavene hodnoty minut

```



```

{
    count_sec = 60;
    count_minute--;
}

if(count_minute <= 0)//odecitani z nastavene hodnoty minut
{
    count_minute = 60;
    count_hour--;
}

if(TL_long == TRUE && TL_short == FALSE && TLAC == CLOSE &&
TLAC_DIR == INPUT) //prepinani barev LED, podle delky stisku
{
    count_color++;

    if(count_color >= COLOR_WAIT) // doba mezi prepnutimi
    {
        if(index == 7 )
        {
            c_index = 0;
            index = 0;
        }

        count_color = 0;

        ++index;

        if(index >= pocet_stupnu+1)
            {// dle poctu stupnu se zastavi
                index = pocet_stupnu;
            }

    }
    rezimy_led();
}
else
{
    count_color = 0;
}

}

/***** end timer0 *****/

```

```

/*****stav time*****/
void time() // rozhodnutí mezi kratkym a dlouhym stiskem kratky-
> nesviti, dlouhy -> hodnota pro timer
{

    if(stav == STAV_ROZSVIT)
    {
        /*pro stav rozsvit*/
        index = 1;
        TL_long = TRUE;
        TL_short = FALSE;
        while (TLAC == CLOSE)
        {

            color(c_index);

        }
        count_sec = 60;
        count_minute = casy_m[index];
        count_hour = casy_h[index];
    }

    /*pro stav TIME*/
    if(stav == STAV_TIME)
    {
        blik_en = FALSE;
        count_color = 0;
        index = 7;
        color(index);

        while(TLAC == CLOSE )
        {
            doba_stisku();
        }

        if(TL_long == TRUE)
        {
            stav = STAV_SVITI;
            count_minute = casy_m[index];
            count_hour = casy_h[index];
            count_sec = 60;
            count_pom = pulse_sec;
            rezimy_led();
        }
    }
}

```

```

        if(TL_short == TRUE)
        {
            stav = STAV_NESVITI;
        }

    }

}

/*****end stav time*****/
void sviti()
{
    blik_en = ON;
    if(TLAC == CLOSE)
    {
        stav = STAV_TIME;
        blik_en = OFF;
    }
    watch();
}

/*****end time *****/
/*****stav rozsvit*****/
void rozsvit() //rozsviti a preda rizeni
{
    RELE = ON;
    time();
    stav = STAV_SVITI;
}

/*****funce blikaj*****/
void blik(int bperiod,int bpulse ,int a,int b,bool en)
{
    //funkce zajistuje blikani danou freq, stridou, dve barvy

    if(en == ON )
    {
        ++count_blik;
        if(count_blik >= bperiod) //generovani
        {

```

```

        count_blik = 0;

    }
    if(count_blik <= bpulse)
    {
        color(a);

    }
    if(count_blik >= bpulse )
    {
        color(b);
    }

}

}

/*****end blikej*****/
/*****stav servis*****/
void servis()
{
    if(blik_en == OFF)
    {
        vypocet(FREQ_BLIK_SET,DUTY_BLIK_SET);
        c_index = nastaveni+1; // barva nastaveni
        sec_icolor = 0; // bude se stridat
        count_pom = 0; // nulovani pomocne promene pro osetreni
                        zakmitu tlacitka
    }
    blik_en = ON; // povoleni blikani

    if(NASTAV_TLAC == 0)
    {

        if(count_pom >= TL_WAIT)
        {
            nastaveni = ++nastaveni; // po stisku TL2 se zvysi
                                    index nastaveni
            if(nastaveni >= POCET_NASTAVENI ) // rotace
                                    nastaveni dokola
            {
                nastaveni = 0;
            }
        }
    }
}

```

```

        while(NASTAV_TLAC == CLOSE) // cekani na uvolneni
                                           TL2
        {
            index = nastaveni+1;
            c_index = index;
        }

    }
}
else
{
    count_pom = 0;
}

if(TLAC == CLOSE && TLAC_DIR == INPUT)// navrat do nesviti
{

    blik_en = ON;
    sec_icolor = 0;
    index = 0;
    write_EE(EESET,nastaveni);
    stav = STAV_INIT; // restart a nacteni parametru
}

}
/***** end servis *****/
/*****funkce volby barvy , parametr *****/
void color(int color_index)
{
    switch(color_index)
    {
        case 0:
            NIC();
            break;

        case 1:
            FIALOVA();
            break;

        case 2:
            MODRA();
    }
}

```

```

        break;

        case 3:
            TYRKYSOVA();
        break;

        case 4:
            ZELENA();
        break;

        case 5:
            ZLUTA();
        break;

        case 6:
            CERVENA();
        break;

        case 7:
            BILA();
        break;

    }

}

/*****end barva *****/
/*****funkce zjisti den
stisku*****/
void doba_stisku()
{
    if(count_TL <= TL_LONG && count_TL >= TL_WAIT )
    {
        TL_long = OFF;
        TL_short = ON;

    }
    if(count_TL >= TL_LONG)
    {
        TL_long = ON;
        TL_short = OFF;

    }
}

```

```

}
/*****end doba stisku*****/
/*****funkce WATCH*****/
void watch()
{
    if( (count_hour*255)+count_minute <= (255*casy_h[index-
1])+casy_m[index-1] )// hlidani podkroceni intervalu a zmena barvy
    {
        --index;
        rezimy_led();

    }

    if(count_minute <= casy_m[index-1]+1 && count_sec <=
ALERT_TIME_30 && index == 1)// zrychlení blikání na konci posledního
intervalu
    {

        if(count_sec <= ALERT_TIME_10)
        {
            vypocet(FREQ_BLIK_A2,DUTY_BLIK_A2); //2. upozneni
        }
        else
        {
            vypocet(FREQ_BLIK_A1,DUTY_BLIK_A1); //1. upozneni
        }
    }
    else
    {

        vypocet(FREQ_BLIK_STD,DUTY_BLIK_STD);

    }
    if((count_minute == 0 && count_hour == 0)|| index == 0)
    {

        stav = STAV_NESVITI;

    }
}

```

```

}
/*****end watch*****/
/*****funkce vypocet *****/
void vypocet(int freq, int percent)
{
    period = pulse_sec/(freq
    pulse = (percent*period)/100;
    //return;
}
/*****end vypocet*****/
/*****funkce zapis do EEPROM*****/
void write_EE(int8_t eadr, int8_t edata ) // funkce pro zapis do
EEPROM
{
    EEADRL = eadr;
    EEDATL = edata;
    CFGS = 0;
    EEPGD = 0;
    WREN = 1;
    GIE = 0;
    //vyzadovana sekvence z DS
    EECON2 = 0x55;
    EECON2 = 0xAA;
    WR = 1;
    //****konec sekv.
    GIE = 1;
    WREN = 0;
    while(WR == 1) //cekání na kocnec zapisu do EE
    {

    }

}

/*****end write EEPROM*****/
/*****funkce cteni z eeprom*****/
int8_t read_EE(int8_t eadr) //funkce pro cteni z EEPROM, vrací
                                dat z EE
{
    EEADRL = eadr;
    CFGS = 0;

```



```

    EEPGD = 0;
    RD = 1;
    return EEDATL;

}
/*****end read eeprom *****/
/*****funkce rezim LED *****/
void rezimy_led()
{
    if(pocet_stupnu == 6)
    {
        c_index = index;
        color(c_index);
    }
    if(pocet_stupnu == 2)
    {
        if(index == 2)
        {
            c_index = COLOR_LONG;
            color(c_index);
            sec_icolor = KD_SEC_COLOR;
        }
        if(index == 1)
        {
            c_index = COLOR_SHORT;
            color(c_index);
            sec_icolor = KD_SEC_COLOR;
        }
        if(index == 0)
        {
            c_index = 0;
            color(c_index);
        }
    }
}
/*****end rezim led*****/

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////  END CODE  ///////////////////////////////////
/////////////////////////////////////////////////////////////////

```

Hlavičkový soubor - main.h

```
#include <xc.h>
#define _XTAL_FREQ 4000000

//define DEMO      //Pro jiné zapojení LED
#define ON          1
#define OFF         0

#define INPUT       1
#define OUTPUT      0

#define OPEN        1
#define CLOSE       0

#define TRUE        1
#define FALSE       0

// CONFIG1
#pragma config FOSC = INTOSC
// Oscillator Selection (INTOSC oscillator: I/O function on CLKIN
pin)
#pragma config WDTE = OFF
// Watchdog Timer Enable (WDT disabled)
#pragma config PWRTE = ON
// Power-up Timer Enable (PWRT enabled)
#pragma config MCLRE = OFF
// MCLR Pin Function Select (MCLR/VPP pin function is digital input)
#pragma config CP = OFF
// Flash Program Memory Code Protection (Program memory code
protection is disabled)
#pragma config CPD = OFF          // Data Memory Code Protection (Data
memory code protection is disabled)
#pragma config BOREN = OFF       // Brown-out Reset Enable (Brown-out
Reset disabled)
#pragma config CLKOUTEN = 1     // Clock Out Enable (CLKOUT function
is disabled. I/O or oscillator function on the CLKOUT pin)
#pragma config IESO = OFF       // Internal/External Switchover
(Internal/External Switchover mode is disabled)
#pragma config FCMEN = ON       // Fail-Safe Clock Monitor Enable
(Fail-Safe Clock Monitor is enabled)
```

```

// CONFIG2
#pragma config WRT = OFF
    // Flash Memory Self-Write Protection (Write protection off)
#pragma config PLLEN = OFF
    // PLL Enable (4x PLL disabled)
#pragma config STVREN = OFF
    // Stack Overflow/Underflow Reset Enable (Stack Overflow or
    Underflow will not cause a Reset)
#pragma config BORV = LO
    // Brown-out Reset Voltage Selection (Brown-out Reset Voltage
    (Vbor), low trip point selected.)
#pragma config LVP = OFF
    // Low-Voltage Programming Enable (Low-voltage programming enabled)

////////////////////////////////////
// HW I/O definitions
////////////////////////////////////

#ifdef DEMO //prepinani podeltypu led
#define RED            RA2
#define BLUE           RA0
#else

#define RED            RA0
#define BLUE           RA2

#endif

#define GREEN          RA1
#define NASTAV_TLAC    RA3
#define TLAC           RA4
#define RELE           RA5

#define TLAC_DIR       TRISA4
////////////////////////////////////
// HW makra
////////////////////////////////////

#define PWR_OFF()      {TLAC_DIR = 0; TLAC = 0;}
#define NIC()          {RED = 0; GREEN = 0; BLUE = 0;} //0
#define FIALOVA()      {RED = 1; BLUE = 1; GREEN = 0;} //1
#define MODRA()        {BLUE = 1; RED = 0; GREEN = 0;} //2
#define TYRKYSOVA()    {BLUE = 1; GREEN = 1; RED = 0;} //3
#define ZELENA()       {GREEN = 1; RED = 0; BLUE = 0;} //4
#define ZLUTA()        {GREEN = 1; RED = 1; BLUE = 0;} //5

```

```

#define CERVENA()    {RED = 1;BLUE = 0; GREEN = 0;} //6
#define BILA()       {RED = 1; GREEN = 1; BLUE = 1;} //7

////////////////////////////////////
// parametry
////////////////////////////////////

#define TMRO_REFILL      108 //obnovovací hodnota pro 400 Hz

#define TL_LONG           400// doba dlouheho stisku sec/400
#define PULSE_PER_SEC     400// pocet preruseni za sec.
#define COLOR_WAIT        200// doba mezi prepnutimi barev sec/400
#define TL_WAIT           30// osetreni zakmitu kontaktu tlacitka
#define POCET_NASTAVENI    6 //pocet moznych nastavení
#define COLOR_SHORT        6 //barva pro kratky cas ve dvou casovem
režimu
#define COLOR_LONG         2 //barva pro dlouhy cas ve dvou casovem
režimu
#define KD_SEC_COLOR       4 //barva stridající se s barvou ve dvou
časovém režimu
#define OFF_COLOR          0 //barva v pohotovostnit režimu
#define ALERT_TIME_30      30 //1 vystraha zbyvajiciho času (sec)
#define ALERT_TIME_10      10 //2 vystraha zbyvajiciho času (sec)
#define FREQ_BLIK_STD       2 //frekvence standartního režimu
#define DUTY_BLIK_STD      50 //strida standartního režimu

#define FREQ_BLIK_SET      10//frekvence při nastavovani
#define DUTY_BLIK_SET      70 //strida při nastavovani

#define FREQ_BLIK_A1       4 //frekvence při 1 upozorneni
#define DUTY_BLIK_A1      50//strida při 1 upozorneni

#define FREQ_BLIK_A2       8 //frekvence při 2 upozorneni
#define DUTY_BLIK_A2      30 //strida při 2 upozorneni

#define EESET              0xA0 //adresa nastaveni
#define OFF_COLOR_ADR      0xF0 //adresa barvy nestviti

```